JOSÉ AUGUSTO AFONSO

Estimativa de Parâmetros de Qualidade de Serviço no Acesso a Recursos da Internet

Dissertação para Mestrado em Informática

Escola de Engenharia
Departamento de Informática
UNIVERSIDADE DO MINHO

Braga, 1996

JOSÉ AUGUSTO AFONSO
(Bolseiro da Junta Nacional de Investigação Científica e Tecnológica)

Estimativa de Parâmetros de Qualidade de Serviço no Acesso a Recursos da Internet

Tese submetida à Universidade do Minho para a obtenção do grau de mestre em Informática, área de especialização de Sistemas Distribuídos, Comunicações por Computador e Arquitectura de Computadores, elaborada sob a orientação do Professor Doutor Vasco Luís Barbosa de Freitas.

Escola de Engenharia
Departamento de Informática
UNIVERSIDADE DO MINHO

Braga, 1996

Resumo

A qualidade de serviço no acesso a recursos da Internet muitas vezes é comprometida pela distancia entre clientes e servidores. Quando o servidor é popular a situação se agrava ainda mais.

A replicação de recursos, seja estática ou dinâmica, torna-se cada vez mais necessária para a solução destes problemas, uma vez que o número de utilizadores da Internet cresce a um ritmo frenético.

Tão importante quanto a replicação é a existência de um método de selecção entre servidores que permita informar a um cliente qual servidor tem probabilidade de lhe oferecer um dado recurso em menor tempo. A escolha adequada contribui ainda para distribuir o tráfego na Internet com mais eficiência, beneficiando indirectamente outros utilizadores.

A tarefa de selecção não é fácil, porém, porque o estado da rede varia continuamente ao longo do tempo. E como o método de selecção deve basear-se em estimativas feitas anteriormente para cada servidor, é desejável que o processo de obtenção destas estimativas não gere muito tráfego adicional, nem deixe os clientes à espera por muito tempo.

Nesta tese é desenvolvido um método de selecção entre servidores baseado na técnica de *passive probing*. O critério de selecção é a qualidade de serviço esperada de cada servidor, expressa em função do tempo de resposta e da disponibilidade.

Com o uso de *passive probing*, os resultados de conexões anteriores feitas por clientes locais a cada servidor são utilizadas para estimar o tempo de resposta para a próxima conexão, evitando a inserção de tráfego na rede para fazer as medições.

O cálculo das estimativas baseia-se principalmente em amostras do tempo de estabelecimento de conexão recolhidas de conexões passadas. O MSS da conexão também é considerado.

O método proposto é comparado com outros que apresentam finalidades semelhantes, mostrando um desempenho superior em muitos casos. Os resultados experimentais indicam que o método definido neste trabalho é capaz de reduzir o tempo de resposta médio em mais de 50 %, quando comparado com o método de selecção aleatório.

A maior demanda para um método de selecção entre servidores actualmente advém do World-Wide Web. Assim, as medições neste trabalho foram feitas utilizando-se o protocolo HTTP, embora a expansão a outros protocolos seja sempre considerada.

Palavras chave: Internet, World-Wide Web, Qualidade de Serviço, Selecção entre Servidores, *Passive Probing*.

Abstract

The quality of service on the access of Internet resources is prejudiced many times by the distance between clients and servers. The situation is even worse when the server is popular.

The replication of resources, static or dynamic, becomes each time more necessary to solve these problems, because the number of users of Internet is growing very quickly.

As important as replication is the presence of a server selection method that informs to a client the server that will probably deliver a given resource in less time. The right choice also contributes to a more efficient distribution of the Internet's traffic, with indirectly benefits to the other users.

The server selection task is not easy, though, because the network state varies continually along the time. And as the server selection method should rely on prior estimates made to each server, it's desirable that the measurements necessary to calculate those estimates don't introduce too much traffic on the network, nor let the clients wait for a long time.

On this thesis, a server selection method based on the passive probing technique is developed. The selection criterion is the quality of service expected from each server, expressed as a function of the response time and the availability.

With passive probing, the results of anterior connections to each server, made by local clients, is used to estimate the response time for the next connection, avoiding the insertion of traffic on the network to make the measurements.

The calculation of the estimates is mainly based on samples of the connection establishing time recorded from past connections. The connection's MSS is also considered.

The proposed method is compared with others that present similar purposes, showing better performance in many cases. Experimental results suggest that the server selection method defined on this work is able to reduce the average response time in more than 50 %, when compared with the random selection mechanism.

The higher demand for a server selection method today comes from the World-Wide Web. So, the measurements on this work were made using the HTTP protocol, even though the expansion to other protocols has been always considered.

Keywords: Internet, World-Wide Web, Quality of Service, Server Selection, Passive Probing.

Agradecimentos

Gostaria de agradecer às seguintes pessoas e instituições pela sua contribuição na realização desta tese:

- Professor Vasco Freitas pela orientação e disponibilização de recursos;
- JNICT e Gabinete de Gestão do PRAXIS XXI, por proporcionarem condições indispensáveis à execução deste trabalho;
- Colegas de Mestrado pela companhia, em especial à Cláudia Reis pela parceria na resolução dos problemas;
- Grupo de Comunicações por Computador, pelo auxílio e esclarecimento de dúvidas;
- Universidade do Minho, em particular ao Departamento de Informática e ao Centro de Informática, pelos meios postos à disposição.

O trabalho desenvolvido nesta tese foi efectuado no âmbito do Programa PRAXIS XXI.

Índice

Lista de Figuras

Lista de Tabelas

Glossário

ASCII	American Standard Code for Information Interchange.
CERN	Conseil Européen pour la Recherche Nucléaire
CIUM	Centro de Informática da Universidade do Minho.
DARPA	Defense Advanced Research Projects Agency
DIT	Directory Information Tree.
DNS	Domain Name System.
DUI	Directory User Interface.
FTP	File Transfer Protocol.
HTML	Hypertext Markup Language.
HTTP	Hypertext Transfer Protocol.
IETF	Internet Engineering Task Force.
IP	Internet Protocol.
MIME	Multipurpose Internet Mail Extensions.
MSS	Maximum Segment Size.
NCSA	National Center for Supercomputing Applications
RCCN	Rede da Comunidade Científica Nacional.
RFC	Request for Comments.
SDUM	Serviços de Documentação da Universidade do Minho.
TCP	Transmission Control Protocol
UDP	User Datagram Protocol.
URC	Uniform Resource Characteristics.
URL	Uniform Resource Locator.
URN	Uniform Resource Name.
WAIS	Wide Area Information Servers.
WWW	World-Wide Web.

1 Introdução

1.1 Motivação

Em um sistema de informação distribuído como o World-Wide Web [1], a utilização de um único servidor para fornecer um dado recurso pode apresentar problemas que degradam a qualidade de serviço (QoS) oferecida aos clientes. Um dos problemas consiste na demora para obtenção do recurso a que os clientes distanciados do servidor são submetidos, devido ao facto de que os pacotes trocados pelos dois *hosts* passam por várias redes intermediárias antes de atingir o seu destino. Outro problema é a sobrecarga a que um servidor popular é submetido para atender aos pedidos de uma vasta audiência, que pode ocasionar uma deterioração na qualidade de serviço mesmo para clientes situados nas proximidades do servidor.

A solução para estes problemas consiste na colocação de réplicas dos recursos em outros servidores distribuídos pela Internet. A utilização de servidores localizados nas proximidades dos clientes, além da distribuição da carga entre os servidores e a redução do tempo de resposta, possibilita a diminuição do tráfego de pacotes em redes distanciadas dos clientes, contribuindo para a redução da carga global na Internet.

Entretanto, a replicação de recursos não é suficiente para garantir uma qualidade de serviço melhor aos clientes, nem a diminuição do tráfego global na Internet, pois os clientes poderão optar erradamente por servidores mais distantes.

Assim, para que os benefícios da replicação sejam integralmente alcançados, é necessário um mecanismo de selecção que permita informar ao cliente qual servidor está mais próximo de si, dentre um conjunto de servidores disponíveis para o recurso em causa.

1.2 Objectivos

Existem diversos critérios que podem ser utilizados para a definição da proximidade entre o cliente e o servidor. A distância geográfica talvez seja o mais óbvio, embora o número de *hops* se aplique melhor às redes de computadores. Em [2] os autores preferem evitar o uso de *backbones* na comunicação entre os *hosts*, mesmo que isso implique num tempo de resposta maior. Nosso ponto de vista é que, para fins de selecção, a proximidade de um servidor deve ser avaliada primeiramente em função da estimativa da qualidade de serviço, através dos parâmetros relacionados com o tempo necessário à obtenção do recurso; mesmo que isso implique, em alguns

casos, na selecção de um servidor mais distante em termos geográficos ou de número de *hops*, pois o que importa para o usuário é obter o recurso no menor tempo possível.

O objectivo principal deste trabalho consiste na elaboração de um método de selecção que seja capaz de reduzir o tempo de resposta necessário à obtenção de recursos, comparando-se com o método de selecção aleatória.

Para além desse objectivo, outras características são desejáveis para o método adoptado:

- O processo utilizado para o cálculo das estimativas deve introduzir um mínimo de carga adicional na rede.

- O tempo que o cliente tem que esperar para que a selecção seja feita deve ser pequeno.

1.3 Organização desta Tese

No capítulo 2 apresenta-se uma descrição sucinta das tecnologias relacionadas com este trabalho, focando principalmente os pontos de maior interesse.

No capítulo 3 a abordagem torna-se mais restrita ao assunto desta tese. Fazem-se os primeiros testes, bem como a análise de trabalhos relacionados. E retiram-se as primeiras conclusões, que servem de base para a decisão do caminho a seguir.

O capítulo 4 aprofunda-se na técnica de *passive probing*. As alternativas são avaliadas e define-se os factores a considerar na estimativa da qualidade de serviço. São feitas medições, e os resultados obtidos servem para comprovação da eficácia do modelo proposto, bem como a comparação com outros modelos.

No capítulo 5 apresentam-se os aspectos relacionados com a implementação do modelo proposto, incluindo a exposição dos problemas enfrentados e suas soluções.

Finalmente, no capítulo 6 fazem-se as considerações finais com respeito a esta tese, discutem-se os resultados obtidos, e propõem-se caminhos a seguir para a continuação deste trabalho.

2 Enquadramento Geral

2.1 A Internet

A Internet é resultado do trabalho iniciado na década de 60 pelo DARPA (Defense Advanced Research Projects Agency), do Departamento de Defesa dos EUA, que definiu um conjunto de especificações para a comunicação entre computadores de diferentes redes, conhecido como família de protocolos TCP/IP [3].

Inicialmente voltada para comunicações militares do governo americano, a utilização da Internet foi posteriormente alargada aos meios académicos. A partir daí, suas infra-estruturas expandiram-se gradualmente pelo planeta, servindo a aplicações como correio electrónico, grupos de discussão, transferência de ficheiros e terminal virtual.

Com o surgimento do World-Wide Web, em 1990, a Internet passou a atrair o interesse crescente de novos utilizadores fora do meio académico, motivando a sua exploração comercial e a ampliação dos investimentos em infra-estrutura.

A família de protocolos TCP/IP, na qual baseia-se a Internet, é organizada em quatro camadas. Acima dos protocolos de acesso à rede situa-se a camada internet, com o protocolo IP (Internet Protocol), que implementa os procedimentos necessários à comunicação entre dois *hosts* quaisquer na Internet, incluindo a transição de pacotes entre redes e o encaminhamento destes até o destino.

Acima da camada Internet situa-se a camada de transporte, cuja função é providenciar a comunicação de dados entre dois processos em diferentes hosts. Dentre os protocolos desta camada, os mais utilizados são:

- UDP (User Datagram Protocol) - fornece as funcionalidades mínimas da camada. Tem a vantagem de introduzir um *overhead* pequeno nas comunicações, mas por outro lado não garante a sua fiabilidade.

- TCP (Transmission Control Protocol) [4] - É um protocolo orientado à conexão que garante uma comunicação de dados fiável à camada superior, implementando o controlo de erros e de fluxo de dados, e assegurando que os pacotes chegam na sequência correcta ao destino.

Na camada superior localiza-se o protocolo de nível aplicação, como o HTTP, que define a forma de acesso a recursos disponíveis na Internet.

2.2 O World-Wide Web

O WWW foi idealizado por investigadores do CERN, o Laboratório Europeu de Física de Partículas, em Genebra, na Suíça. A ideia básica de sua utilização baseia-se no paradigma do hipertexto. Após ser obtido um documento de hipertexto, a selecção de um *link* neste faz com que o cliente busque outro documento (que pode ser procedente de um outro servidor). Este documento, por sua vez, pode conter outros *links*, e assim o processo de navegação continua. O WWW também permite que o utilizador envie um texto ao servidor (como numa busca), em vez de seguir um *link*. O servidor analisa o texto e devolve uma resposta de acordo com seu conteúdo.

O WWW ganhou popularidade com o surgimento do Mosaic, um cliente com interface gráfica desenvolvido no NCSA (National Center for Supercomputing Applications), nos EUA, que possibilitou a visualização de textos e gráficos numa mesma página WWW, e facilitou a navegação pela utilização do rato.

O World-Wide Web fundamenta-se em três componentes básicos:

- O URL (Uniform Resource Locator) [5]
- O HTML (Hypertext Markup Language) [6] [7]
- O HTTP (Hypertext Transfer Protocol) [8] [9]

O URL é um identificador universal na Internet que permite a localização de um recurso (e definição do protocolo de acesso) de maneira simples. A sintaxe do URL permite a referência não só a recursos que utilizem o HTTP, mas também a recursos disponíveis através de outros protocolos, como o FTP, o NNTP, o Gopher e o WAIS.

O HTTP foi idealizado para possibilitar a transferência dos recursos de modo simples e eficiente. Apesar do nome, o HTTP não se destina apenas à transferência de hipertexto na forma de documentos HTML. O formato utilizado pode ser qualquer outro, possibilitando a transferência de textos simples, imagens, sons, ficheiros compactados, etc. A enorme popularidade do WWW provocou um aumento exponencial na utilização do HTTP. O tráfego HTTP já é bem maior do que o gerado por qualquer outro protocolo na Internet, e continua crescendo [10].

O HTML é a linguagem padrão no WWW para a criação de páginas de hipertexto. Foi concebida para ser suficientemente simples e legível para permitir a criação de documentos tanto manualmente quanto automaticamente. Novas versões do HTML ampliaram as capacidades disponíveis, incluindo suporte para tabelas, *frames*, simbologia matemática, etc.

O WWW continua em expansão, ampliando seus recursos e incorporando novas tecnologias, como a linguagem Java e a realidade virtual. Com isso, os browsers WWW tornam-se cada vez mais complexos, e incorporam cada vez mais recursos multimédia, que exigem a transferência de grande quantidade de informação, o que faz com que o tráfego na Internet aumente ainda mais.

2.2.1 O HTTP

O HTTP é um protocolo de nível aplicação concebido para ser utilizado no World-Wide Web. O HTTP é normalmente implementado sobre o TCP, utilizando para os servidores, por *default*, a porta 80. É um protocolo simples, sem estado, que se baseia no paradigma pedido/resposta. O cliente estabelece uma conexão com o servidor e faz o pedido. O servidor processa o pedido, retorna a resposta, e fecha a conexão.

A figura 1 apresenta a sequência de pacotes trocados entre o cliente e o servidor em uma conexão HTTP típica. Normalmente o pedido feito pelo cliente cabe em um único pacote, podendo haver excepções. Por outro lado, a resposta do servidor geralmente se estende por mais de um pacote, mas também pode caber em apenas um pacote (quando o recurso pedido não é transmitido, ou seu tamanho é muito pequeno).

O pedido do HTTP é composto basicamente pelo método (que define o tipo de acção a realizar pelo servidor), o URL que especifica o recurso, e a versão do protocolo utilizada pelo cliente. A resposta do servidor contém em primeiro lugar uma linha de *status*, que inclui a versão do protocolo utilizada pelo servidor, um código de 3 dígitos que informa ao cliente sobre o resultado do pedido, e um texto complementar destinado ao usuário (opcional).

Tanto o pedido como a resposta podem conter mensagens do tipo MIME contendo pares <*Header*: Valor>. A seguir é introduzido o conteúdo (caso exista), no campo denominado Entity-Body.

Existem diferentes categorias de mensagens associadas ao protocolo HTTP: mensagens gerais, incluindo a data (Date) de envio do respectivo envio ou resposta; e mensagens associadas ao conteúdo, como o seu tamanho (Content-Length), o formato (Content-Type), ou a data da última modificação (Last-Modified). Existem também mensagens associadas somente ao pedido, como informação sobre o cliente (User-Agent), ou somente à resposta, como informação sobre o servidor (Server);

Existem também diversos métodos definidos no protocolo HTTP, sendo que o mais utilizado é o GET, que requisita o recurso referenciado pelo URL. No caso de já existir uma cópia do documento desejado em *cache*, é normal utilizar-se um GET condicional, que instrui o servidor para enviar o documento apenas se este tiver sido modificado desde a data do último acesso. Isso é feito através da inclusão da mensagem "If-Modified-Since" com essa data no pedido.

O método POST é normalmente utilizado para o envio de informação ao servidor, como no caso da submissão de um *form* [6] pelo usuário. Porém, o método GET também pode ser utilizado pelo cliente para a submissão de informação, que é incluída no URL, neste caso.

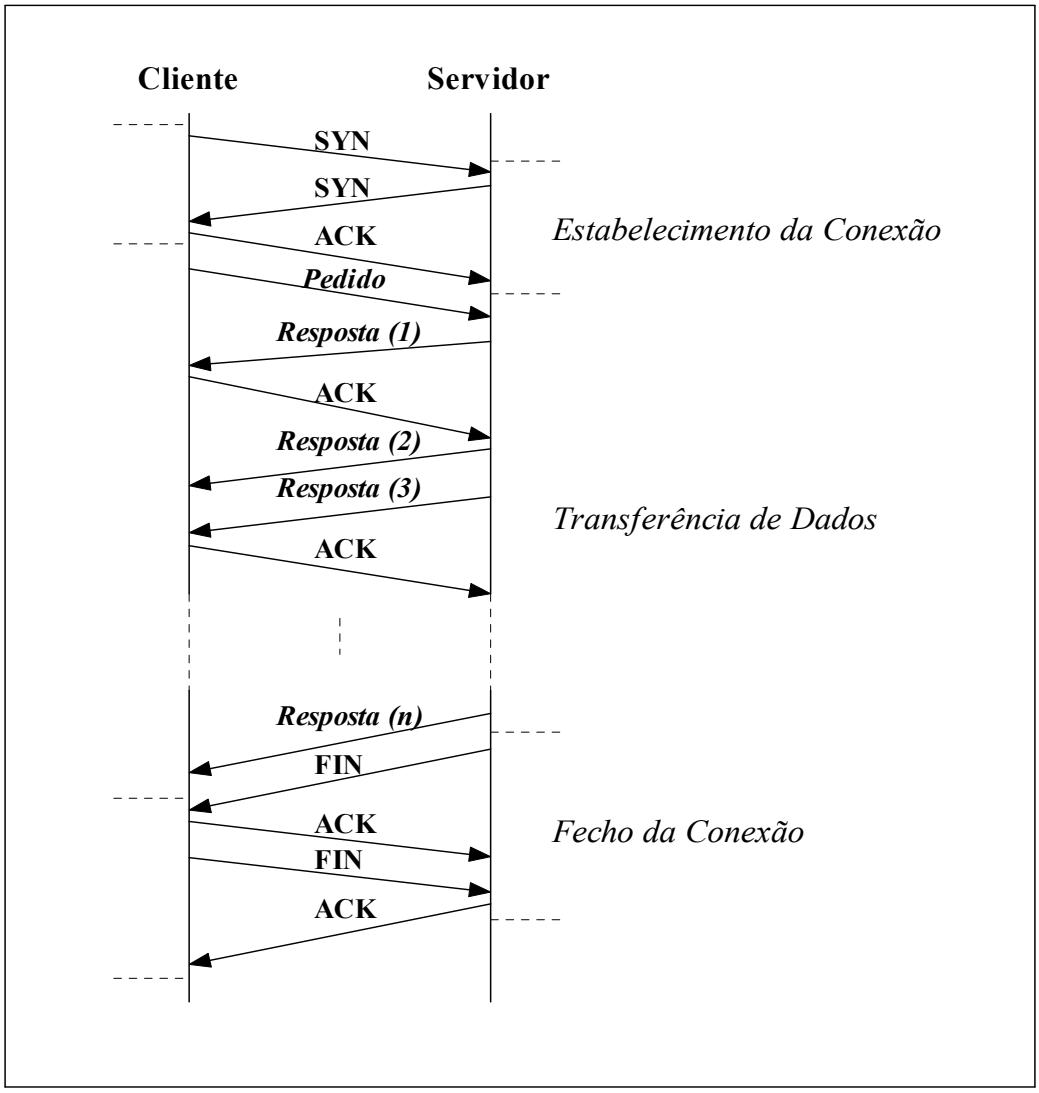


Figura 1: Intercâmbio de Pacotes em uma Conexão HTTP Típica.

O URL utilizado no pedido pode ter dois formatos diferentes. O URL absoluto é utilizada somente quando o pedido é feito a um servidor *proxy*. Um exemplo da linha de pedido neste caso:

GET http://www.uminho.pt/UM/um.html HTTP/1.0

Quando o pedido é enviado directamente ao servidor, descarta-se a informação do protocolo e do nome do servidor. A linha de pedido correspondente ao exemplo anterior ficaria da seguinte forma:

GET /UM/um.html HTTP/1.0

A versão 1.1 do HTTP [11] proporciona uma extensão dos recursos do protocolo. Entre outras características, esta versão inclui novos métodos, mecanismos

de controlo de *caching* de documentos mais elaborados, e suporte à utilização de conexões persistentes.

2.2.2 *Proxies* WWW

Um servidor *proxy* [12] actua como intermediário na comunicação entre clientes e servidores, sendo normalmente utilizado por clientes em redes isoladas, cujo acesso à Internet é feito através de uma máquina *firewall*.

O *proxy* actua como servidor para o cliente, recebendo o seu pedido. Caso tenha o recurso desejado em *cache*, o *proxy* envia a resposta. Caso contrário, faz o pedido ao servidor original do recurso, passando a actuar como cliente, e repassa a resposta ao cliente que lhe fez o pedido.

O *proxy* WWW comunica-se com os seus clientes utilizando o protocolo HTTP, mas pode ser capaz de satisfazer pedidos de recursos disponíveis através de outros protocolos, como Gopher, WAIS, e FTP.

O *caching* de documentos feito pelo *proxy* torna a sua utilização benéfica mesmo para clientes que possuam acesso directo à Internet, e desta forma não necessitem de usar o *proxy* como intermediário.

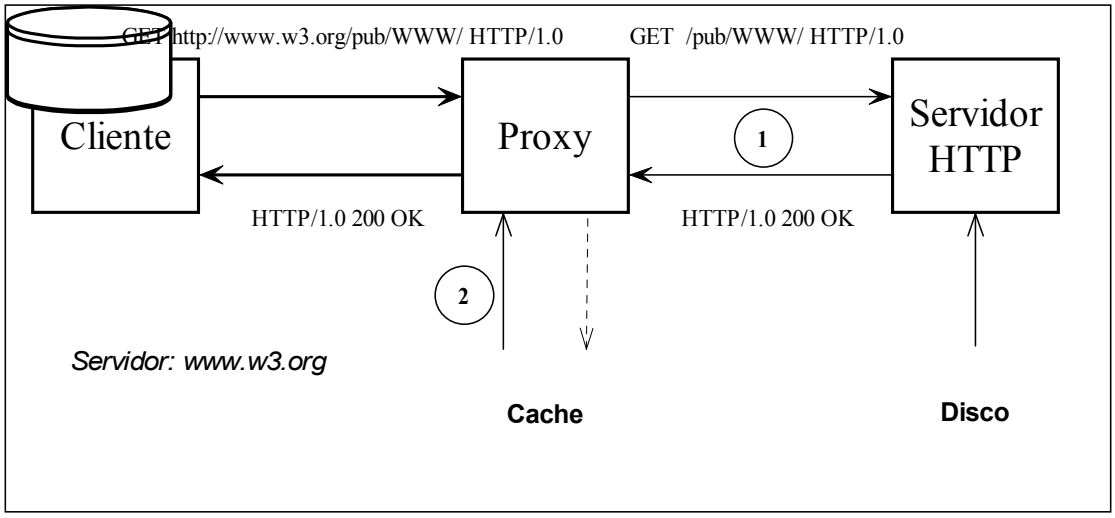


Figura 2: Exemplo da Obtenção de um Documento através de um Proxy.

O cliente faz o pedido ao servidor *proxy* especificando o URL absoluto. O *proxy* refaz o pedido ao servidor HTTP usando um URL relativo, na qual o protocolo e o nome do servidor foram retirados (são óbvios para o servidor). O tipo de URL utilizado permite diferenciar entre pedidos feitos a *proxies* e servidores (tal distinção é necessária, pois um *proxy* pode também actuar como um servidor normal na mesma porta TCP). Dependendo do recurso, este pode ou não ser armazenado em *cache* pelo *proxy*.

A figura 2 apresenta um exemplo da utilização de um proxy para se obter um documento de um servidor HTTP (no caso, www.w3.org). O cliente faz o pedido ao proxy, que pode agir, basicamente, de duas maneiras diferentes:

1. Se o documento não estiver em *cache*, refaz o pedido ao servidor. Este, por sua vez, normalmente busca o documento em disco e o envia ao proxy, que pode armazenar o documento em *cache*, se for apropriado.

2. Se o documento já estiver na *cache* do proxy, este o envia directamente ao cliente.

No caso do URL especificar outro protocolo suportado pelo *proxy*, este se encarrega da tradução da informação do HTTP para o protocolo em questão, e vice-versa. A comunicação do *proxy* com o cliente é feita usando-se o HTTP, enquanto que a comunicação com o servidor é feita utilizando o outro protocolo. Isso também permite que clientes simples que só implementam o HTTP possam ter acesso a recursos disponíveis através de outros protocolos.

2.2.3 Cache

Para evitar a transferência de um documento repetidas vezes de um servidor remoto para um cliente, os *browsers* de WWW normalmente implementam uma *cache* local no cliente. Entretanto, as *caches* dos clientes locais não se comunicam, de forma que um cliente é obrigado a buscar o documento desejado ao servidor, mesmo que um de seus vizinhos o possua em *cache*. O uso de um servidor *proxy* pelos clientes locais ajuda a resolver esse problema, pois os clientes passam a partilhar uma *cache* comum. Porém, como o espaço em disco, tanto na *cache* do cliente, como na do *proxy*, é limitado, os documentos menos populares acabam sendo descartados.

Para aumentar a quantidade de documentos disponíveis em *cache*, há diversos trabalhos que tem como objectivo possibilitar que servidores *proxy* partilhem suas *caches* entre si e com seus clientes. [13]. Diversas questões se apresentam com essa estratégia, por exemplo:

- Como localizar servidores que possuam o documento em *cache*?

- Havendo mais de um servidor disponível, qual escolher?

- Como assegurar a consistência da informação em *cache*?

Normalmente, quem toma a decisão sobre os documentos a armazenar em *cache* é o cliente. Alguns autores consideram vantajoso que a tomada de decisão seja feita por parte do servidor, que disporia de melhores meios para decidir onde e quando replicar os seus documentos, de forma a minimizar o tráfego na rede e distribuir a carga entre os servidores [14].

Estudos sobre o padrão de acesso de documentos em *sites* no World-Wide Web mostram que a maior parte dos acessos remotos é feito a um pequeno conjunto

de documentos do *site* [15]. De facto, quanto mais popular é um *site*, maior é a tendência para a disparidade no número de acessos a seus documentos. Assim, ao disponibilizar-se os documentos mais populares de um servidor em *proxies* localizados nas proximidades de seus clientes, consegue-se diminuir bastante a carga no servidor e reduzir o tráfego na rede, sem que seja necessário ocupar uma quantidade de memória muito grande na *cache* dos servidores *proxies*.

A implementação de mecanismos avançados de *caching* de documentos, como os referenciados anteriormente, não exclui a necessidade de um método de selecção entre servidores eficiente; antes pelo contrário.

2.3 Localização do Recurso

2.3.1 Resolução URN2URC

Um recurso disponível na Internet possui um URL associado, pelo qual é identificado e referido. Caso a localização do recurso se altere, a referência torna-se inválida. Da mesma forma, a existência de cópias do recurso em outras localizações não é conhecida a partir do URL.

A expectativa, porém, é que o URN (Uniform Resource Name) [16] [17] venha a ser utilizado, juntamente com o URC (Uniform Resource Characteristics) [18]. O URN proporciona uma referência ao recurso independente da localização, enquanto que o URC contém um ou mais URLs que apontam para o recurso na rede, além de meta-informação associada ao recurso ou aos URLs (título, versão, formato, etc). A migração para a utilização de URNs e URCs depende da implantação de um serviço de resolução URN2URC [19] eficiente.

A obtenção de um recurso na Internet a partir de um URN é feito após um primeiro passo que consiste na obtenção de um URC associado. Os URLs contidas no URC podem então ser automaticamente enviadas ao processo de selecção entre servidores, que escolhe um URL com base nas estimativas de qualidade de serviço, sem que seja necessária a intervenção do usuário. Em seguida, procede-se o acesso ao recurso, tal como é feito actualmente.

2.3.2 Alternativas

Além da resolução URN2URC, existem diversas outras formas de se obter uma lista de servidores na Internet que disponibilizam o mesmo recurso.

- Cada vez é mais comum o aparecimento de *mirror sites* no World-Wide Web. Sua localização normalmente é disponível a partir da Home Page, tanto do *site* principal como de seus *mirrors*. Exemplos de sites com diversos mirrors são o Tucows [20] e o Linux Documentation Project [21].

- Uma lista de servidores FTP que possuem um dado ficheiro disponível para *download* pode ser obtida fazendo-se uma query a um servidor Archie [22],

especificando na query o nome do ficheiro, ou parte deste. Outros serviços semelhantes estão disponíveis na Internet.

- Diversos *sites* no WWW contém bases de dados com informação referente à localização de ficheiros, fornecendo listas que incluem não só servidores FTP, mas também servidores HTTP, como por exemplo o SHAREWARE.COM [23].

- Para obter uma lista de servidores que possuam cópia de um determinado documento em *cache*, os autores em [14] utilizam uma técnica que consiste em perguntar ao servidor primário (que contém o documento original). O servidor mais próximo do cliente passa a oferecer automaticamente o documento dali em diante.

2.4 Outros Protocolos de Interesse

2.4.1 O DNS

Antes que um *host* possa contactar outro *host* na Internet, precisa conhecer o seu endereço IP. No caso do WWW, normalmente o cliente obtém primeiro o nome do servidor, tendo que fazer uma *query* ao DNS (Domain Name System) [24, 25] para obter o seu endereço.

O DNS possui uma base de dados distribuída por diversos servidores de nomes, com informação associada a nomes de domínio. Os domínios são organizados na estrutura de uma árvore, inspirada na hierarquia organizacional. As diferentes zonas do espaço de domínios são armazenadas em diferentes servidores de nomes, possuindo estes autoridade sobre aquela zona. Um servidor de nomes também pode fazer o *cache* temporário de informação de outros servidores, de forma a agilizar o processo de obtenção de informação no DNS. Caso um servidor de nomes não possua a informação desejada pode indicar um conjunto alternativo de servidores com capacidade de resposta.

O nome de um domínio identifica um nó na árvore de DNS. Cada nó tem um conjunto de informação (que pode estar vazio) composto por *Resource Records* (RR). Existem diversos tipos de *Resource Records*, como por exemplo:

- Tipo A, valor 1 - O endereço de um *host*. No caso da Internet, um endereço IP.

- Tipo NS, valor 2 - O nome de um servidor de nomes autorizado para o domínio.

- Tipo CNAME, valor 5 - O nome canónico de um domínio.

- Tipo HINFO, valor 13 - Informação sobre a CPU e o sistema operativo de um *host*.

- Tipo MX, valor 15 - Um *host* com capacidade para encaminhar mensagens de *mail* para o domínio em causa.

As mensagens do DNS na Internet podem ser transmitidas usando-se o UDP ou o TCP. As actividades de transferência de zona devem usar o TCP devido à necessidade de um transporte fiável. Já as queries utilizam normalmente o UDP, que proporciona um menor *overhead* e melhor desempenho. Os servidores no DNS utilizam a porta 53 em ambos os casos.

As mensagens do protocolo DNS são divididas em 5 secções, sendo que algumas podem estar ausentes em certos casos:

- *Header* - Está sempre presente, e inclui campos que especificam se a mensagem é uma *query* ou resposta, se as demais secções estão presentes, etc.

- *Question* - Contém a questão feita ao DNS. Aparece tanto na *query* como na resposta.

- *Answer* - Lista de RRs respondendo à query.

- *Authority* - Lista de RRs apontando para servidores de nomes com autoridade para responder à query.

- *Additional* - Lista de RRs com informação adicional relacionada com a *query*.

2.4.2 O FTP

O FTP (File Transfer Protocol) [26] é um protocolo de nível aplicação amplamente utilizado na Internet para a transferência de ficheiros entre hosts. O FTP requer a autenticação do cliente, pelo envio de *login* e *password* ao servidor, antes que a transferência de dados seja efectuada, embora muitos servidores aceitem *login* anónimo para permitir a transferência de ficheiros de acesso público.

O FTP opera sobre o TCP, assim como o HTTP. Diferentemente deste, o FTP necessita do estabelecimento de duas conexões entre o cliente e o servidor para a transferência de um ficheiro: uma conexão de controlo e uma conexão de dados. A conexão de dados é terminada após a transferência de um ficheiro, embora a conexão de controlo possa ficar aberta, possibilitando o estabelecimento de conexões de dados subsequentes.

O servidor FTP aguarda pedidos de estabelecimento da conexão de controlo na porta 21, por *default*. O formato utilizado para a transferência de informação na conexão de controlo do FTP é o mesmo formato especificado para o Telnet. A conexão de controlo é utilizada para a autenticação do cliente, passagem de comandos (get <ficheiro>, por exemplo), e comunicação da porta a ser utilizada na conexão de dados.

O estabelecimento da conexão de dados, normalmente iniciado pelo servidor, utiliza a porta 20 (FTP-DATA) no servidor e uma porta que não esteja sendo usada no cliente. Entretanto, este método pode não funcionar através de um *firewall*, pois alguns rejeitam conexões iniciadas do exterior. Por esse motivo, *browsers* WWW como o Netscape [27] utilizam um método conhecido como *passive* FTP para buscar ficheiros em servidores FTP [28]. Neste caso, a conexão de dados também é iniciada

pelo cliente. Em vez de se usar a porta 20 do servidor, utiliza-se uma porta alta, definida através da conexão de controlo.

2.5 O Tcpdump

O tcpdump [29] é um programa que permite monitorar o tráfego que passa em determinado ponto da rede. Para isso, o programa deve ser executado em um *host* que possua interface com a rede no ponto desejado.

O tcpdump faz a leitura dos *headers* de pacotes que transitam pela rede e cujas características satisfazem a uma expressão booleana definida pelo usuário. Caso nenhuma expressão seja definida, todos os pacotes são lidos.

A expressão é composta por primitivas que permitem a selecção de pacotes com base nos protocolos (TCP, UDP, etc), *hosts*, redes e portas (de origem, de destino, ou qualquer). As primitivas são agrupadas utilizando-se os operadores booleanos AND, OR e NOT. Definido um protocolo, também é possível comparar o conteúdo numérico em posições específicas do pacote com valores desejados, permitindo uma maior flexibilidade na selecção de pacotes, baseada nos valores de campos dos *headers* dos protocolos.

Algumas opções da linha de comando do tcpdump relevantes para este trabalho são:

- *-n* Não converte os endereços numéricos dos pacotes em nomes (para a conversão o tcpdump utiliza o DNS).

- *-sN* Extrai os primeiros *N* bytes do pacote. O valor default é 68. Um valor pequeno pode truncar informação relevante em alguns protocolos. Já um valor grande aumenta o tempo de processamento e pode causar a perda de pacotes.

- *-x* Adicionalmente, apresenta o conteúdo do pacote (excepto os bytes do nível de *link* de dados) em hexadecimal.

O formato da informação apresentado pelo tcpdump para os pacotes recolhidos varia de acordo com o protocolo. Para o TCP, o formato utilizado é o seguinte:

timestamp host_org **.** *port_org* **>** *host_dst* **.** *port_dst* **:**
flags seq_n ack window urg options

- *timestamp* - Horário em que o pacote foi detectado pelo tcpdump. A precisão depende do *clock* da máquina.

- *host_org* - *Host* de origem do pacote, representado pelo seu nome (ou endereço IP, no caso da opção *-n* do tcpdump ser utilizada).

- *port_org* - Porta de origem do pacote.

- *host_dst* - *Host* de destino do pacote, representado de forma idêntica à de *host_org*.

- *port_dst* - Porta de destino do pacote.

- *flags* - Normalmente possui um dos seguintes valores: "S" (SYN), "P" (PUSH), "F" (FIN), "R" (RST) ou "." (nenhuma). A *flag* "P" pode vir só ou acompanhada de outra.

- *seq_n* - Números de sequência de dados abrangidos pelo pacote, com o seguinte formato: *primeiro* **:** *último* **(***número de bytes***)**.

- *ack* - Número de sequência de dados esperado.

- *window* - Tamanho da janela de recepção de dados disponível no emissor.

- *urg* - se estiver presente indica que a *flag* URG está activada.

- *options* - Dados do campo de opções do TCP, como o MSS.

Os campos de host e porta de origem e de destino, *timestamp* e *flags* estão sempre presentes. Os outros dependem do pacote.

3 Abordagem Inicial

3.1 Parâmetros de Qualidade de Serviço

Nesta tese são considerados dois factores que afectam a qualidade de serviço: o tempo de resposta e a disponibilidade do servidor, ambos do ponto de vista do cliente. A escolha do servidor apropriado é feita com base numa estimativa desses dois parâmetros, tomando como base medições anteriores feitas para cada servidor da lista. Diversas variáveis que podem ser utilizadas para estimar o tempo de resposta e a disponibilidade de um servidor são discutidas e os resultados obtidos para cada uma são comparados.

Além do tempo de resposta e disponibilidade, existem outros parâmetros de qualidade de serviço associados aos recursos que podem aplicar-se aos servidores na Internet.

- *Integridade dos dados (data completeness)* - se a informação disponível em um dado servidor está completa.

- *Actualidade dos dados (data up-to-dateness)* - se a informação disponível em um dado servidor está actualizada.

Cabe ao processo encarregado da gestão de réplicas fornecer informação adicional associada a cada servidor que indique as diferenças entre as cópias (caso existam). Por exemplo, o formato do URC prevê a inclusão de informação diversa associada a cada URL, como a versão do documento; que também pode ser utilizada como critério de selecção.

3.2 Comportamento dos Tempos de Resposta

O tempo de resposta é aqui definido como o tempo decorrido desde que o cliente envia o pedido de conexão ao servidor até que a transmissão de dados entre ambos termina. Para o caso do HTTP, o tempo de resposta é a soma to tempo de estabelecimento de conexão (referido neste artigo como tempo de conexão), e o tempo de transferência de dados (referido aqui como tempo de transferência).

Os tempos de resposta de um servidor típico na Internet apresentam alta variabilidade, como pode-se observar na figura 3, que mostra medições consecutivas

do tempo de resposta de um servidor para o mesmo documento. O intervalo entre as medições é de 1 hora.

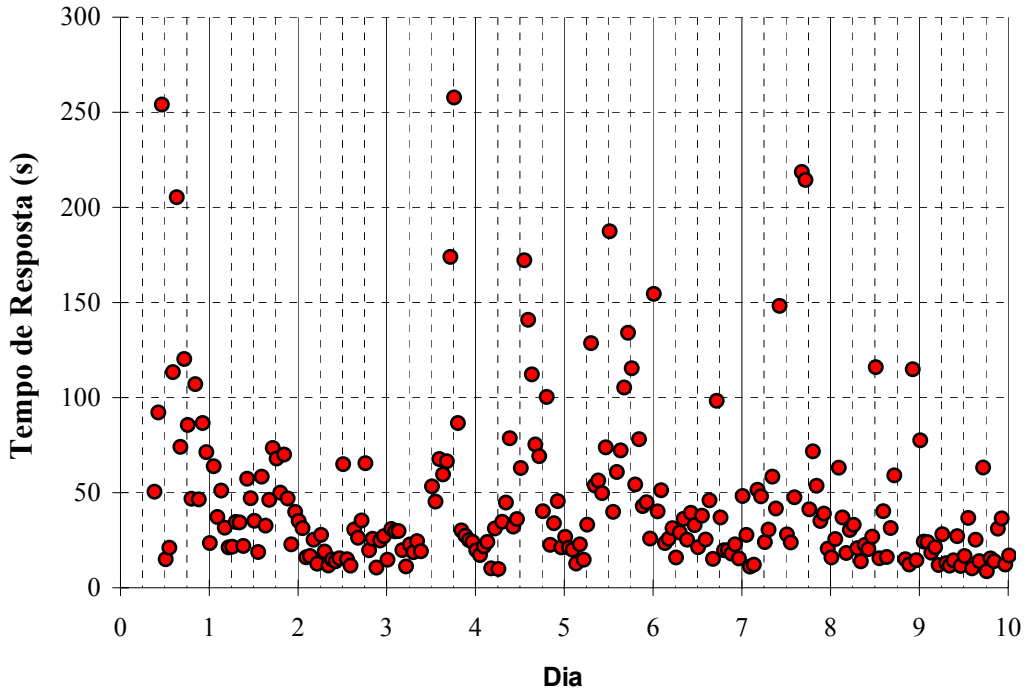


Figura 3: Tempos de Resposta de um Servidor ao Longo do Tempo.

De facto, a taxa de transferência de dados pode mudar abruptamente mesmo durante uma conexão, com reflexos no tempo de resposta. A figura 4 mostra sucessivas medições do tempo de transferência em função do número de *bytes* recebidos de um servidor. Notam-se, por vezes, grandes variações na taxa de transferência, mesmo entre a chegada de pacotes consecutivos.

Tal comportamento dos tempos de transferência pode ser explicado por variações bruscas da carga das redes situadas no caminho entre o cliente e o servidor, provocando congestionamento de pacotes nos *routers* e causando atrasos no envio, ou mesmo o descarte de pacotes.

Como a carga na rede e a congestão tem como principal causa o tráfego competitivo, o tempo de resposta é influenciado pela da acção de um grande número de hosts que utilizam as redes no caminho entre o servidor e o cliente. Como consequência, torna-se quase impossível prever com um alto grau de certeza o tempo de resposta que se vai obter de um servidor genérico na Internet, pois as condições existentes no instante em que é feita a estimativa podem alterar-se de um momento para o outro de forma imprevisível. Como a estimativa do tempo de resposta de um servidor possui intrinsecamente uma margem de erro elevada, o processo de selecção do servidor (que se baseia nas estimativas feitas para cada um dos servidores) é incapaz de fazer sempre a melhor escolha, qualquer que seja o método utilizado para fazer a previsão.

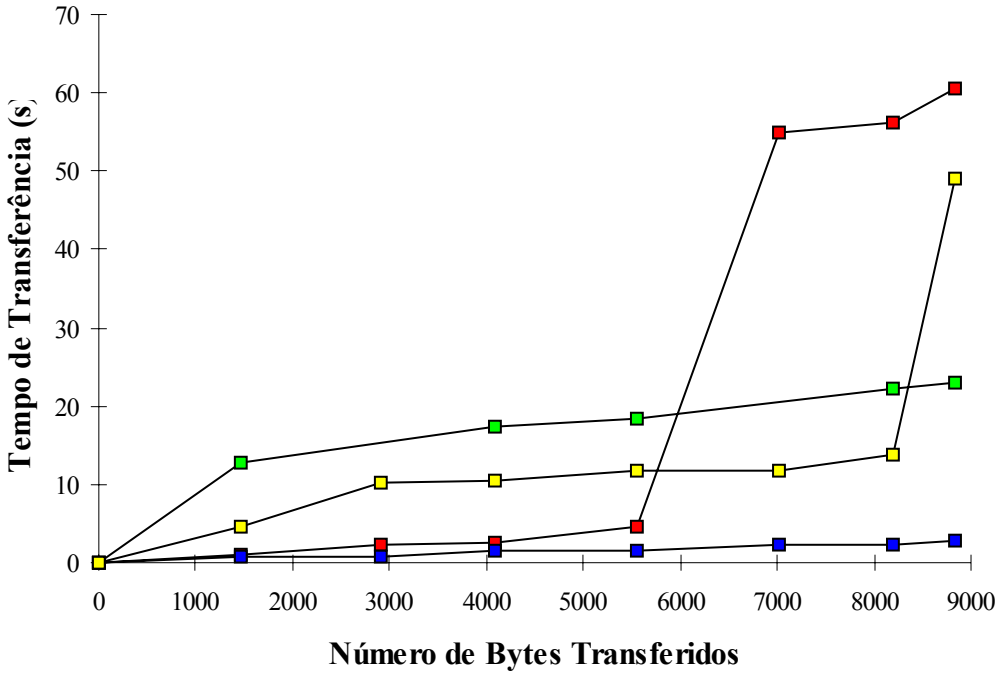


Figura 4: Tempos de Transferência de um Servidor em Função do Número de Bytes Transferidos em Medições Consecutivas.

Assim, o objectivo principal do trabalho pode ser melhor definido como a elaboração de um método de previsão capaz fazer uma boas escolhas na maior parte dos acessos, de modo que o tempo de resposta necessário para a obtenção de um conjunto de documentos seja inferior, em média, ao tempo que seria gasto ao se utilizar um mecanismo de selecção aleatório. Ou seja, apesar de a escolha feita poder não ser sempre a melhor quando examinados os casos individualmente, que se consiga no conjunto de todas as medições um ganho na qualidade de serviço oferecida aos clientes.

3.3 Estimativas e Métodos de Selecção

3.3.1 Critérios Básicos de Estimativa de Proximidade

Para se fazer a selecção entre servidores que disponibilizam um mesmo recurso, diversos critérios podem ser utilizados.

Alguns autores baseiam-se na distância geográfica entre o cliente e o servidor, considerando, assim, a proximidade física entre os hosts para fazer a selecção. Uma estimativa precisa da localização de um host poderia ser feita consultando-se o *Resource Record* GPOS [30] do DNS, mas este raramente é usado. Em [31], o critério da distância geográfica é utilizado, tanto para a definição de onde são

colocadas as réplicas de documentos, como para seleccionar o servidor mais próximo de um cliente.

Entretanto, o uso da distância geográfica como critério de selecção não é satisfatório, pois não diminui eficazmente nem o tráfego na rede nem o tempo de resposta. O facto de um servidor estar mais próximo de um cliente do que outro não significa que o caminho a percorrer pelos pacotes seja menor. Por exemplo, os pacotes que transitam entre um cliente localizado no campus de Gualtar da Universidade do Minho (em Braga) e o servidor www.tsc.uvigo.pt (em Vigo, na Espanha) passam por Lisboa e Madrid, duas cidades que estão muito mais afastadas. A escolha de um servidor em Lisboa, em vez de um servidor em Vigo, diminuiria bastante o caminho a percorrer, o número de *hops* entre o cliente e o servidor e, consequentemente, o tráfego gerado na rede.

Quanto ao tempo de resposta, também sofre influência do número de *hops* entre os *hosts*, dependendo ainda de outros factores, como a largura de banda e taxa de utilização dos *links*, e da carga no servidor. Os factores ligados à topologia da Internet e à carga na rede variam com o tempo, os primeiros a longo prazo, e os últimos a curto prazo, fazendo com que o servidor mais adequado à selecção também varie (a distância geográfica entre *hosts*, porém, é constante). Portanto, o método de selecção entre servidores deve ser dinâmico, não devendo basear-se em um critério estático.

Outro critério de distância que produz melhores resultados é o número de *hops* entre os *hosts*. Esta informação pode ser obtida utilizando-se o traceroute [32]. O principal benefício que se pode obter com a utilização deste critério para a selecção entre servidores é a redução do tráfego na rede, já que o servidor escolhido é aquele para o qual o trânsito de pacotes se faz pelo menor número de *links*.

Porém, o número de hops não é uma boa estimativa do tempo de resposta, visto que este depende de outros factores enunciados anteriormente. Também não se pode afirmar que a diminuição do tráfego pelo uso desse critério seja sempre a mais apropriada, pois sua utilização pode levar à escolha de um caminho congestionado em detrimento de outro subutilizado, bastando este último ser ligeiramente mais longo. Portanto, o uso do critério do número de *hops* não garante uma distribuição eficiente da carga na Internet.

Em [33], os autores mediram o número de hops e o *round-trip time* entre um cliente e um conjunto de servidores na Internet. O *round-trip time* foi medido utilizando-se as mensagens de *echo request* e *echo reply* do ICMP (Internet Control Message Protocol), sendo esta medida conhecida também pelo nome de tempo de ping, devido ao nome do *software* que se costuma utilizar para se fazer este tipo de medições, normalmente incluído em distribuições do TCP/IP.

Os resultados mostram que a distribuição de frequência do número de *hops* aproxima-se da distribuição normal, com média $\bar{x}$ = 16,6 e desvio padrão σ_x = 4,0. Resultados semelhantes foram obtidos em [2].

Já a distribuição de frequências do *round-trip time* apresenta um padrão assimétrico, em que a maior parte das amostras localiza-se abaixo da média (apresentando similaridades com as distribuições exponencial e Gamma). Como consequência, o valor obtido para a mediana (125 ms) é muito menor do que para a média (241 ms).

Dadas as diferenças entre as distribuições de número de *hops* e *round-trip times*, a correlação entre ambos, calculada pelos autores, é naturalmente baixa (10 %).

Esse tipo de distribuição de frequências assimétrico é característico não só de *round-trip times*, como também de outras medições de tempo na Internet, como é o caso do tempo de resposta e do tempo de conexão, de acordo com as nossas medições. Em [34] os autores encontraram resultados semelhantes para a latência no acesso a servidores HTTP, definida em seu trabalho como sendo o tempo necessário para a resolução do nome do servidor, estabelecimento de conexão e envio do pedido. Portanto, o uso do número de hops para a estimativa de qualquer desses tempos não deve oferecer uma boa acurácia.

Prosseguindo o seu trabalho [33], os autores mediram, periodicamente, para um conjunto de 10 servidores, o *round-trip time* 5 vezes (com o ping) e, imediatamente a seguir, o tempo de resposta para documentos de diferentes tamanhos. Com esses dados, simularam o uso de diversos métodos para a selecção entre servidores:

1. Estático, baseado na distância geográfica.

2. Estático, baseado no número de *hops*.

3. Dinâmico, baseado na selecção aleatória.

4. Dinâmico, baseado na média de 1 a 5 medições do *round-trip time*.

Os melhores resultados (menores tempos de resposta médios) foram obtidos com o uso do *round-trip time*, enquanto que os piores foram obtidos usando-se a distância geográfica. Os resultados obtidos com o número de *hops* também foram maus, em alguns casos piores do que a simples utilização da selecção aleatória.

Esses resultados indicam que para se obter uma boa estimativa do tempo de resposta deve-se utilizar um método dinâmico, como o apresentado no item 4. Existem outras métricas temporais que podem ser utilizadas, porém, bem como formas alternativas de se fazer as medições, dando origem a diversos métodos de selecção. A seguir é feita uma análise desses métodos.

3.3.2 Métodos Temporais

3.3.2.1 Estimativa Instantânea

A estimativa instantânea , conforme definido aqui, é calculada com base em medições concentradas em um curto período. As medições normalmente são feitas em resposta ao pedido de selecção, para uso imediato. Existem muitos métodos nesta categoria, variando basicamente quanto à métrica utilizada.

Um método utilizado por muitos autores, com pequenas variações, consiste na medição simultânea do *round-trip time* para um conjunto de servidores, imediatamente antes de o pedido ser feito. O servidor que responde primeiro, ou seja, aquele cuja valor do *round-trip time* medido for menor, é o escolhido.

A implementação do SONAR [35] utiliza este método para ordenar uma lista de hosts (representados por endereços IP) de acordo com a proximidade destes com o servidor SONAR. A medição do *round-trip time* também é feita utilizando-se as mensagens de *echo* do ICMP. Para diminuir o tráfego decorrente do envio de diversas mensagens para um mesmo *host* (em *queries* deferentes), o valor do *round-trip time* é armazenado em *cache* durante algum tempo.

Um método semelhante é utilizado em [13], neste caso para a selecção dentre *caches* organizadas hierarquicamente. Quando uma *cache* recebe o pedido de um URL que não possui, ela executa uma *remote procedure call* para todas as suas *siblings* (*caches* no mesmo nível hierárquico) e suas *parents* (*caches* no nível hierárquico superior) à procura do recurso. A *cache* original obtém o recurso da primeira *cache* contactada a responder com um "Hit".

Esse método é similar ao que foi denominado Dyn 1 em [36], que utiliza uma medição única do *round-trip time* (com o ping) para cada servidor. Os autores defendem que uma medição mais completa do estado da rede pode conduzir a uma melhor estimativa do tempo de resposta. Com esse objectivo, resolveram experimentar duas medidas.

a) Utilizar a média de um conjunto de *round-trip times*, em vez de apenas uma medição para cada *host*, pois, devido à grande variância apresentada pelas medições do *round-trip time* para um mesmo *host* (mesmo em curtos períodos de tempo), uma única amostra do *round-trip time* pode conduzir a uma estimativa incorrecta.

b) Estimar a largura de banda disponível para cada host imediatamente antes da selecção, pois a influência da largura de banda no tempo de resposta pode ser significativa na transferência de documentos maiores.

Para a estimativa da largura de banda disponível , os autores elaboraram duas ferramentas de *probing*: BPROBE, para estimar a largura de banda básica em uma conexão, e CPROBE, para estimar o grau de congestão de uma conexão. Ambas, tal como o *round-trip time*, foram implementadas utilizando-se as mensagens de *echo* do ICMP.

Com BPROBE envia-se uma sequência de pacotes ao destinatário e verifica-se a resposta obtida. Baseado nos tempos de entre-chegadas e tamanhos dos pacotes, tenta-se estimar a largura de banda básica (ou seja, a menor largura de banda dentre os *links* no caminho entre os dois *hosts*). A largura de banda básica limita a taxa de transferência máxima entre os dois *hosts*. Os dados têm que ser enviados a alta velocidade, para que haja *queuing* no *router* do *bottleneck link*. Diversos factores (como o tráfego competitivo e a perda de pacotes) influenciam as medições, podendo levar a resultados imprecisos. Para diminuir o erro na estimativa faz-se o envio de um grande número de pacotes de diferentes tamanhos e uma filtragem dos resultados, para descartar as medições incorrectas.

Com CPROBE envia-se uma sequência de pacotes a uma alta taxa de transmissão. Na resposta, mede-se o tempo decorrido entre a recepção do primeiro e do último pacote. Com isso, consegue-se medir a presença de tráfego competitivo no *bottleneck link*. Para diminuir a influência da perda ou reordenação de pacotes pelo caminho são feitas 10 medições, das quais são aproveitadas 4.

A associação da média de um conjunto de medições do *round-trip time* com a estimativa da largura de banda, foi denominada Predicted TT (*predicted transfer time*), sendo calculada da seguinte forma:

$$Predicted\ TT = k_1\ RTT + k_2\ \frac{S_{doc}}{B_{avail}} \quad (1)$$

Em que *RTT* é a média dos *round-trip times*, B_{avail} é a largura de banda disponível, calculada utilizando-se BPROBE e CPROBE, S_{doc} é o tamanho do documento, k_1 e k_2 são coeficientes que ajustam o peso de cada uma das componentes na estimativa do tempo de resposta.

Para testar o método de selecção baseado no Predicted TT, os autores recolheram periodicamente de um conjunto de servidores a seguinte informação: 5 medições do *round-trip time*, a largura de banda do *bottleneck link* (com BPROBE), a utilização do *link* (com CPROBE) e o tempo de resposta. Foram utilizados documentos grandes, de 100 KB a 1 MB, para avaliar a influência da largura de banda disponível nos resultados.

Com esses dados, os autores simularam a selecção com base nos seguintes critérios:

- Selecção aleatória.
- Número de hops
- Dyn 1 (uma única medição do *round-trip time*)
- Dyn 5 (a média de 5 medições do *round-trip time*)
- Predicted TT

O cálculo da correlação entre tempos de resposta e as estimativas feitas (para os diversos documentos) indica um aumento da acurácia ao se utilizar o Predicted TT, comparando-se com a estimativa feita com a utilização de uma única medição do *round-trip time*.

Quanto ao tempo de resposta médio, os piores resultados foram obtidos com a selecção aleatória, seguida pelo número de *hops*, enquanto Dyn 1 apresentou resultados intermediários. Dyn 5 e Predicted TT apresentaram os melhores resultados, porém, muito semelhantes, apesar de Dyn 5 ser uma estimativa muito mais simples. Os autores concluem com isso que a média dos *round-trip times* já considera em certo grau os efeitos do tráfego na rede sobre a largura de banda disponível.

Os autores também encontraram um número significativo de servidores para os quais a estimativa da largura de banda disponível foi bem maior que a taxa de transferência real. Observou-se que os servidores para os quais isso acontecia eram populares, demonstrando que a carga nos servidores tem grande influência nos tempos de resposta. Nenhuma das métricas baseadas no ping (incluindo-se o Predicted TT) consideram eficazmente a carga no servidor, pois os pacotes do ICMP são processados a um nível mais baixo pelo *host*, e a resposta é enviada logo que possível.

Para estimar a carga nos servidores, os autores utilizaram a técnica de buscar um documento inexistente em servidores WWW. A resposta dos servidores, neste caso, é, basicamente, uma mensagem de erro. Observou-se uma relação entre tempos de resposta maiores e as estimativas erradas da taxa de transferência. Os autores concluem, porém, que fazer uma conexão com cada servidor apenas para buscar um documento de teste não compensa de modo algum os custos envolvidos, embora seja salientada a necessidade de um método de medição da carga nos servidores que apresente menores custos, para ser incluído na estimativa dos tempos de resposta.

Outro factor que pode causar erros na estimativa é que o tratamento dado aos pacotes do ping na rede pode ser diferente daquele dado aos pacotes do protocolo usado pelo servidor. Como exemplo, em [35] os autores referem o facto de algumas redes utilizarem um *firewall* que rejeita mensagens de *echo request* do ICMP vindos do exterior (por razões de segurança) embora pacotes de outros protocolos, como o HTTP e o FTP, transitem sem problemas. Neste caso, a estimativa do tempo de resposta baseada no ping para os servidores atrás do *firewall* não poderia ser feita, pois estes pareceriam indisponíveis.

A medição do estado da rede imediatamente antes da conexão apresenta um compromisso entre a precisão das estimativas e os custos envolvidos. O uso de um único ping é a alternativa de menor custo, porém, mede a latência com grandes hipóteses de erro, devido à grande variância das medidas e da elevada porcentagem de perda de pacotes típicas da Internet, além de não oferecer muita informação sobre a largura de banda disponível ou a carga no servidor.

Por outro lado, uma estimativa mais acurada, como a obtida com o uso do Predicted TT, requer a inserção de uma grande quantidade de tráfego na rede a cada vez que uma selecção é feita, além de que o tempo necessário para a obtenção da estimativa aumenta.

3.3.2.2 Estimativa pela Média

O efeito do tráfego competitivo sobre a latência e a largura de banda disponível é muito variável. Em [36], os autores suspeitam que, para documentos grandes, o próprio tempo de transferência excede o período de validade da estimativa, apesar de o tempo de transferência médio em causa ser da ordem de 30 s apenas. Portanto, a estimativa instantânea do tempo de resposta só tem valor efectivo em um período muito curto a contar daquele instante.

Por outro lado, se em vez de se fazer uma estimativa instantânea para um dado servidor, procurar-se obter uma média entre medições feitas em instantes diferentes, e se essa média apresentar-se razoavelmente estável durante um longo período, então pode-se evitar a necessidade de inserção de tráfego na rede para fazer as estimativas toda vez que se deseje fazer a selecção entre servidores.

A selecção baseada na média não vai decidir-se sempre pelo servidor que apresentará menor tempo de resposta, mas é provável que faça uma boa opção. De qualquer modo, a selecção baseada na estimativa instantânea também comete erros devido à grande variabilidade no estado da rede.

Assim, pode-se pensar na eficácia do método de utilizar a média como estimativa, não pelo resultado que pode ser obtido a curto prazo, para uma única conexão, mas pelo resultado global, a médio e longo prazo. O método será eficaz

caso consiga poupar tempo no conjunto dos acessos, de acordo com o objectivo definido anteriormente.

Pode-se evitar a inserção de tráfego na rede de todo com esse método caso se utilize a técnica do *passive probing*. Com esta técnica, a estimativa do tempo de resposta de um servidor é obtida a partir de acessos anteriores ao mesmo. O uso do *passive probing* para a estimativa da qualidade de serviço e selecção entre servidores é analisado no próximo capítulo.

4 Passive Probing

Neste capítulo, a selecção entre servidores baseada no uso do *passive probing* é considerada com maior profundidade, e a comparação com os resultados obtidos utilizando-se outros métodos de selecção é feita. Ao longo da tese, vantagens e desvantagens da utilização da técnica do *passive probing* em diferentes configurações são apresentadas.

4.1 Trabalhos Relacionados

Em [37], o autor utiliza a técnica do *passive probing* para a colecta de informação referente à qualidade de serviço no acesso aos serviços da Directoria X.500. O principal objectivo do autor, entretanto, é fornecer a informação de QoS ao usuário.

A informação, mantida em uma base de dados, provém dos acessos feitos pela DUI (Directory User Interface) utilizada pelo usuário.

Tal como nesta tese, o autor concentra-se em dois aspectos da qualidade de serviço: a disponibilidade dos dados e o tempo de resposta. Os registos na base de dados contém os seguintes campos:

- O objecto base - O nome de uma entrada na DIT (Directory Information Tree).
- Uma média móvel dos tempos de resposta para esse objecto base.
- O resultado da última *query* (sucesso, falha, abandono).
- Um *timestamp* do momento da última *query*.
- Um contador do número de *queries* que tiveram sucesso ou falharam.

A partir da *query* produzida pelo usuário, a DUI consulta a base de dados e produz um aviso ao usuário nas seguintes situações:

- Se a informação normalmente está disponível, mas o tempo de resposta é tipicamente elevado.

- Se a informação não encontra-se disponível no momento da *query*, mas normalmente está disponível.

- Se a informação normalmente não está disponível, não valendo a pena repetir a *query* a curto prazo.

Caso a *query* seja respondida sem demoras, nenhuma mensagem é mostrada ao usuário.

Para diminuir o tamanho da base de dados, as partes da directoria que consistentemente funcionam bem não ficam registadas. Essa opção contrasta com o critério que deveria ser aplicado no caso da selecção entre servidores se fosse preciso limitar o tamanho da base de dados. Neste caso, os registos mais importantes são justamente os correspondentes à melhor qualidade de serviço.

O método descrito pelo autor é bastante simples, entretanto, produz resultados úteis, segundo o mesmo. Porém, a implementação utilizada possui algumas deficiências que comprometem a precisão das estimativas. Como exemplo, todas as queries feitas a um mesmo objecto base são tratadas da mesma forma, apesar de o tempo de resposta variar significativamente de acordo com o tipo de operação realizado. Para as finalidades a que se destina os resultados podem ser satisfatórios, entretanto a selecção entre servidores requer maiores cuidados com a precisão das estimativas.

Em [38] os autores consideram a utilização do *passive probing* para a obtenção de informação de qualidade de serviço, com o objectivo de auxiliar na selecção entre URLs juntamente com outros critérios, como o protocolo de acesso e o formato em que se encontra disponével o recurso. Embora a abordagem utilizada tenha sido superficial (o foco do artigo concentra-se em outros assuntos), as ideias apresentadas serviram de base para a elaboração desta tese.

4.2 Colecta da Informação

A informação obtida com o *passive probing* deve ser armazenada em uma base de dados (que recebeu a denominação de Tabela de QoS) para que esteja disponível quando for necessário. Como novas conexões são feitas a todo momento, e o comportamento da rede é dinâmico, a actualização dos dados na Tabela de QoS deve ser feita continuamente. O primeiro problema com o qual se deparou para o uso *do passive probing* foi como obter a informação necessária à manutenção da Tabela de QoS de modo eficiente.

Seria fácil para um cliente manter uma Tabela de QoS com informação obtida somente de seus próprios acessos. Porém, esta solução é insatisfatória devido à pouca quantidade de informação que um único cliente é capaz de recolher. A informação obtida dos acessos feitos por um cliente pode ser utilizada por todos os clientes locais (embora já não tenha a mesma utilidade para clientes externos), assim, a configuração ideal consiste em uma Tabela de QoS partilhada por todos os clientes locais, em que a informação seja obtida de acessos feitos por um grande número de clientes. Quanto maior o número de clientes fornecendo informação para a Tabela de QoS, maior

número de conexões serão registadas e, portanto, mais completa e actualizada estará a base de dados.

Entretanto, a informação recolhida pelos clientes locais teria que ser processada centralmente, junto à base de dados, obrigando os clientes a ter que se comunicar continuamente com o *host* encarregado da gestão da Tabela de QoS, para enviar a informação recolhida.

Caso os clientes utilizem os serviços de um *proxy*, as conexões com os servidores são feitas por intermédio deste, que pode, desta forma, encarregar-se de colectar a informação referente a cada conexão e ao mesmo tempo gerir a Tabela de QoS. O processo de selecção do servidor pode ser realizado facilmente pelo *proxy* também, já que os dados contidos na Tabela de QoS estão prontamente disponíveis. Com o servidor *proxy* atendendo a pedidos de uma grande quantidade de clientes a Tabela de QoS contará com bastante informação para que a selecção entre servidores possa ser feita de forma eficiente.

Entretanto, existe uma alternativa capaz de tornar a gestão da Tabela de QoS independente tanto de *proxies* como de clientes. Isso é possível através da monitorização do tráfego que passa pela rede [39], seleccionando-se os pacotes que interessem, e extraindo-se destes a informação necessária ao preenchimento da Tabela de QoS. Este método permite inclusive a obtenção mais informação útil à selecção do que os clientes são capazes de fornecer, pois existe a possibilidade de examinar a informação contida nos protocolos de mais baixo nível.

A principal vantagem da monitorização do tráfego, porém, é que os clientes e *proxies* não precisam de sofrem nenhuma modificação para que a colecta de informação de seus acessos seja feita. Além disso, todos os clientes podem contribuir para a Tabela de QoS, mesmo os que não utilizam os serviços de um *proxy*.

4.3 O que medir?

A análise apresentada nesta tese foi feita para o protocolo HTTP, devido à sua enorme popularidade. O HTTP utiliza o TCP (Transmission Control Protocol) para controlar a comunicação entre os hosts, assim como outros protocolos de nível aplicação que requerem um serviço de transporte de dados fiável, por exemplo, o FTP.

Assim, as estimativas de tempo de resposta e disponibilidade baseados no *passive probing* são feitos tendo-se em consideração os diversos componentes de uma conexão TCP. Considerou-se inicialmente duas métricas para o cálculo da estimativa do tempo de resposta de um servidor:

a) O tempo necessário ao estabelecimento de uma conexão TCP (abreviado aqui para tempo de conexão). Esta métrica apresenta alguma semelhança com o *round-trip time* medido pelo ping.

b) Uma função do tempo de resposta para uma conexão e o número de bytes transferidos na mesma. A métrica utilizada deve ser independente do documento de *probing*, mas o tempo de resposta tende a aumentar com o tamanho do documento, havendo a necessidade da normalização.

Na monitorização do tráfego de pacotes, os dados contidos nos *headers* dos protocolos IP e TCP, juntamente com a informação do momento em que o pacote passou pela rede, são suficientes para se obter informação referente ao tempo de conexão, tempo de resposta, número de *bytes* transferidos pelo cliente e pelo servidor. Informação adicional pode ser obtida, se necessário, do *header* do protocolo de nível aplicação, que no caso analisado é o HTTP.

Para facilitar a tarefa de colectar informação dos pacotes que passam pela rede é utilizado um programa chamado tcpdump, que apresenta os *headers* dos pacotes seleccionados que passam pela interface de rede do *host* no qual é executado. Cada pacote recebe um *timestamp*, correspondente ao momento em que foi detectado pelo programa. O critério de selecção dos pacotes é definido pelo utilizador através de uma expressão booleana, na qual se compara os valores do *bytes* contidos em determinadas posições nos pacotes com os valores desejados, possibilitando a filtragem de pacotes de acordo com o endereço IP e a porta do cliente e do servidor, o protocolo utilizado, o tipo de pacote, o endereço da rede, etc.

Para colectar informação a respeito das conexões que utilizem o protocolo HTTP, seleccionam-se os pacotes TCP que utilizem, pelo *host* de origem ou destino, a porta 80 (utilizada por *default* pelo HTTP). Nem todos os pacotes HTTP/TCP precisam ser recolhidos. Para obter-se o tempo de estabelecimento de conexão basta a leitura dos pacotes SYN do TCP. Já para a medição do tempo de resposta necessita-se basicamente dos pacotes SYN e FIN do TCP da parte do cliente. O número de *bytes* transferidos tanto pelo cliente como pelo servidor pode ser obtido do pacote FIN do cliente (ou do pacote FIN do servidor). Juntamente com o *timestamp* registado pelo tcpdump, a informação necessária para ambas as métricas *a* e *b* fica disponível.

4.3.1 Exemplo de Utilização do Tcpdump

A seguir é apresentado um exemplo, com comentários, de uma transacção HTTP tal como é captada pelo tcpdump (comparar com o diagrama apresentado na figura 1). O cliente (alfa.di.uminho.pt) utiliza uma porta aleatória, enquanto que o servidor (zeca.uminho.pt) utiliza a porta 80 (HTTP).

A linha de comando do tcpdump neste caso tem a seguinte forma:

tcpdump -x -s100 tcp and port 80 and host alfa.di.uminho.pt and zeca.uminho.pt

Nos dados em hexadecimal, os 14 *bytes* do protocolo de *link* de dados são descartados, portanto, dos 100 *bytes* definidos pela opção *-s*, apenas os 86 primeiros a partir do *header* do protocolo IP são apresentados.

1) O cliente envia o pedido de estabelecimento de conexão ao servidor (*flag* "S"), e anuncia um MSS de 1460. Caso a não resposta não chegue a tempo, o pacote é retransmitido diversas vezes.

```
18:47:33.702720 alfa.di.uminho.pt.61470 > zeca.uminho.pt.80: S
3262764544:3262764544(0) win 8760 <mss 1460> (DF)
```

```
                4500 002c 9895 4000 fe06 433c c188 1403
                c188 09e6 f01e 0050 c279 d600 0000 0000
                6002 2238 4c0b 0000 0204 05b4 30ec
```

2) O servidor responde aceitando o pedido de conexão. Como não há anúncio do MSS por parte do servidor, considera-se para este o valor *default* de 536. Os dois primeiros pacotes são suficientes para se obter o tempo de conexão, através da diferença entre os *timestamps* (neste caso, cerca de 2 ms). O tempo de conexão neste caso é pequeno porque foram utilizados dois *hosts* locais, mas no contacto com servidores externos o tempo de conexão é muito maior. Para filtrar somente os pacotes do tipo SYN, inclui-se na expressão do tcpdump a primitiva "tcp[13] & 2 != 0".

```
18:47:33.704672 zeca.uminho.pt.80 > alfa.di.uminho.pt.61470: S
2019520000:2019520000(0) ack 3262764545 win 16384
                4500 0028 2872 0000 3c06 b564 c188 09e6
                c188 1403 0050 f01e 785f 6e00 c279 d601
                5012 4000 5f8e 0000 0000 0000 0000
```

3) O cliente reconhece o pacote enviado pelo servidor (ack 1). O tcpdump não escreveu "ack 2019520001" porque da primeira vez que ele identifica uma conexão TCP é apresentado o número de sequência real para o pacote, mas para os pacotes seguintes da mesma conexão ele mostra a diferença entre o número actual e o número inicial, embora este comportamento pode ser alterado pelo o uso da opção *-S* (com letra maiúscula).

```
18:47:33.705648 alfa.di.uminho.pt.61470 > zeca.uminho.pt.80: .
ack 1 win 9112 (DF)
                4500 0028 9896 4000 fe06 433f c188 1403
                c188 09e6 f01e 0050 c279 d601 785f 6e01
                5010 2398 7bf7 0000 785f 6e00 c279
```

4) O cliente envia o pedido do HTTP, que, neste caso, contém 256 *bytes*, cabendo, portanto, inteiramente em um pacote (o MSS da conexão é 536, o menor entre os MSSs anunciados pelos *hosts*). O pacote é enviado com a *flag* “P” activada, para que seja imediatamente processado no destino, em vez de ficar à espera de que o *buffer* fique cheio. O conteúdo do pacote pode ser examinado directamente (em hexadecimal, organizado em grupos de 2 *bytes*). Os primeiros 20 *bytes* compõem o *header* do protocolo IP, enquanto os 20 *bytes* seguintes fazem parte do *header* do TCP. Após esses 40 *bytes* (podem ser mais, caso exista o campo de opções no IP ou no TCP) aparece o *header* do HTTP. Pode-se ler (em ASCII) que o conteúdo da primeira linha do pedido do HTTP é "GET / HTTP/1.0", solicitando o documento "index.html" do directório raiz do servidor (tem o mesmo efeito que "GET /index.html HTTP/1.0"). A seguir aparecem as mensagens do protocolo HTTP, cujo conteúdo é truncado.

```
18:47:33.712480 alfa.di.uminho.pt.61470 > zeca.uminho.pt.80: P
1:266(265) ack 1 win 9112 (DF)
                4500 0131 9897 4000 fe06 4235 c188 1403
                c188 09e6 f01e 0050 c279 d601 785f 6e01
                5018 2398 7959 0000 4745 5420 2f20 4854
                5450 2f31 2e30 0d0a 4966 2d4d 6f64 6966
```

```
                6965 642d 5369 6e63 653a 2054 7565 7364
                6179 2c20 3236
```

5) O servidor inicia o envio da resposta do HTTP. Tal como no pedido, a *flag* "P" é activada. Normalmente, apenas o primeiro e o último pacote da resposta possuem a *flag* "P" activa. Pode-se ler dos *bytes* que a primeira linha da resposta do HTTP é "HTTP/1.0 304 Not modified", que indica que o documento não foi modificado desde a data especificada na mensagem "If-Modified-Since", enviada com o pedido. O documento requisitado não é enviado pelo servidor, apenas o *header* do HTTP, cujo comprimento é de 82 *bytes*.

```
18:47:33.725168 zeca.uminho.pt.80 > alfa.di.uminho.pt.61470: P
1:83(82) ack 266 win 16384
                4500 007a 2875 0000 3c06 b50f c188 09e6
                c188 1403 0050 f01e 785f 6e01 c279 d70a
                5018 4000 53b8 0000 4854 5450 2f31 2e30
                2033 3034 204e 6f74 206d 6f64 6966 6965
                640a 4461 7465 3a20 5468 752c 2030 3420
                4a61 6e20 3139
```

6) O servidor requer que a conexão seja terminada pela sua parte (flag "F" activa). Neste ponto a resposta já foi totalmente enviada. Do campo *ack* retira-se a informação do número de *bytes* enviados pelo cliente, enquanto que do campo *seq_n* obtém-se a informação do número de *bytes* enviados pelo servidor.

```
18:47:33.726144 zeca.uminho.pt.80 > alfa.di.uminho.pt.61470: F
83:83(0) ack 266 win 16384
                4500 0028 2876 0000 3c06 b560 c188 09e6
                c188 1403 0050 f01e 785f 6e53 c279 d70a
                5011 4000 5e33 0000 0000 0000 0000
```

7) O cliente aceita o término da conexão por parte do servidor.

```
18:47:33.727120 alfa.di.uminho.pt.61470 > zeca.uminho.pt.80: .
ack 84 win 9112 (DF)
                4500 0028 9898 4000 fe06 433d c188 1403
                c188 09e6 f01e 0050 c279 d70a 785f 6e54
                5010 2398 7a9b 0000 785f 6e53 c279
```

8) O cliente requer o término da conexão pela sua parte (*flag* "F" activa). O tempo de resposta é obtido pela diferença entre o *timestamp* deste pacote e do primeiro (31 ms). Do campo *seq_n* retira-se a informação do número de *bytes* enviados pelo cliente, enquanto que do campo *ack* obtém-se o número de *bytes* enviados pelo servidor (a mesma informação pode ser obtida do pacote FIN do servidor)

```
18:47:33.733952 alfa.di.uminho.pt.61470 > zeca.uminho.pt.80: F
266:266(0) ack 84 win 9112 (DF)
                4500 0028 9899 4000 fe06 433c c188 1403
                c188 09e6 f01e 0050 c279 d70a 785f 6e54
                5011 2398 7a9a 0000 785f 6e53 c279
```

9) O servidor aceita o término da conexão por parte do cliente.

```
18:47:33.734928 zeca.uminho.pt.80 > alfa.di.uminho.pt.61470: .
ack 267 win 16383
              4500 0028 2877 0000 3c06 b55f c188 09e6
              c188 1403 0050 f01e 785f 6e54 c279 d70b
              5010 3fff 5e33 0000 0000 0000 0000
```

4.3.2 Comparação entre Métricas

Durante a implementação da métrica *b*, baseada na medição do tempo de resposta e do número de bytes transferidos, apareceram alguns problemas relacionados com a sua utilização, expostos a seguir:

- Conexões HTTP utilizando o método POST podem levar a erros na estimativa do tempo de resposta, porque o processamento de tais requisições pelo servidor introduz uma demora adicional, sem relação directa com o tempo necessário à transferência do documento. O mesmo ocorre para as requisições utilizando o método GET que contenham uma *query-string* associada. Caso o erro na estimativas tenha relevância, é necessário descartar o uso destas conexões. Para fazer isso, é necessário recolher os pacotes PUSH do TCP provenientes do cliente para que seja examinado o *header* do pedido do HTTP.

- Conexões HTTP utilizando o método GET com o URL absoluto fornecem uma estimativa incorrecta, pois o destino de tais pacotes é um servidor *proxy*, entretanto o tempo de resposta depende tanto do servidor *proxy* quanto do servidor referenciado pelo URL, bem como da existência ou não do documento na *cache* do *proxy*. Isso não é problemático caso os servidores para os quais se deseja estimar o tempo de resposta não funcionem ao mesmo tempo como *proxies* para os clientes locais na mesma porta.

- Para obter uma boa precisão na estimativa feita com a métrica *b*, é desejável um valor elevado para o número de *bytes* transferidos, pois a transferência de dados é feita em pacotes. Entretanto, grande parte das conexões examinadas apresentou poucos *bytes* transferidos, como no caso das respostas HTTP do tipo "Not Modified", em que o documento requisitado não é transferido por não ter sido modificado desde o último acesso. Mesmo quando um documento é transferido, na maior parte das conexões, o seu tamanho é pequeno, devido ao facto de ser comum no WWW a produção de páginas repletas de pequenas imagens, bem como a prática de dividir o conteúdo de um texto por diversas páginas. Estudos [40] mostram que a mediana do tamanho dos ficheiros transferidos situa-se, na maior parte dos casos, entre 1 e 2 KB.

- Os pacotes RST (RESET) do TCP, provenientes do cliente ou do servidor, também devem ser recolhidos, pois nestes casos a conexão termina sem o envio dos pacotes FIN do TCP em grande parte dos casos. É muito frequente conexões terminarem com RESET, como nos casos em que o utilizador de um *browser* WWW selecciona o botão de *stop* para interromper a recepção de uma página.

- A utilização de conexões HTTP persistentes [40] torna mais difícil a identificação do fim da transferência de um documento e o início da transmissão do seguinte dentro de uma mesma conexão. Entre o término do envio do primeiro documento e o pedido do segundo decorre um intervalo de tempo indeterminado, e não se pode considerar simplesmente o tempo de resposta total e o número de *bytes* transferidos na conexão, neste caso, sem introduzir erros na estimativa.

- Torna-se necessária a definição de uma função que relacione o tempo de resposta e o número de *bytes* transferidos que forneça resultados confiáveis independentemente do número de *bytes* transferidos. Uma função simples seria a divisão do tempo de resposta pelo número de bytes, mas como os dados são transmitidos em pacotes, isso deveria ser levado em consideração. Uma função que fornecesse uma boa precisão na maior parte dos casos seria difícil de se obter, principalmente porque em grande parte das conexões o número de bytes transferidos é muito baixo.

As consequências desses problemas são as seguintes

I. As medições feitas em muitas conexões teriam que ser descartadas, devido à erros ou pouca precisão, reduzindo quantidade de amostras disponíveis ao cálculo das estimativas.

II. O número de pacotes que devem ser recolhidos pelo tcpdump aumenta, e o programa que faz a colecta da informação proveniente de cada conexão torna-se complexo, devido à necessidade de identificação das conexões a serem descartadas e também das conexões que foram abortadas. Além disso, passa-se a exigir maiores recursos computacionais.

III. Pela possível necessidade de se examinar o *header* do HTTP, o programa torna-se dependente da versão, podendo precisar de ser alterado para suportar actualizações do HTTP, ou para o caso da utilização com outros protocolos.

Os problemas apresentados anteriormente levam a optar-se pela a utilização da métrica baseada no tempo de estabelecimento de conexão, que apresenta as seguintes vantagens em relação à outra:

- É independente do documento requisitado ao servidor, ou seja, o número de bytes transferidos tanto pelo servidor quanto pelo cliente não influenciam no valor da métrica.

- Todas as conexões feitas ao servidor podem ser incluídas, independente do método HTTP utilizado, tamanho do documento, do uso de conexões HTTP persistentes ou de eventuais interrupções da conexão.

- Não há necessidade de examinar o *header* do HTTP (ou de qualquer outro protocolo de nível aplicação que seja monitorizado), simplificando a implementação e a extensão do programa a outros protocolos, como o FTP.

- Basta recolher e tratar os pacotes SYN do TCP. Sendo assim, o programa torna-se bem mais simples, e os recursos computacionais envolvidos são bem menores.

4.4 Factores Considerados na Estimativa da QoS

4.4.1 Tempo de Conexão

Os tempos de conexão de um servidor na Internet apresentam grande variância, tal como foi visto para os tempos de resposta. Esse comportamento é de se esperar, já que ambos os tempos são influenciados pelos mesmos factores. A figura 5 apresenta medições consecutivas do tempo de conexão de um servidor típico na Internet, com intervalo de 1 hora entre as medições. Os tempos apresentados anteriormente na figura 3 são provenientes das mesmas conexões.

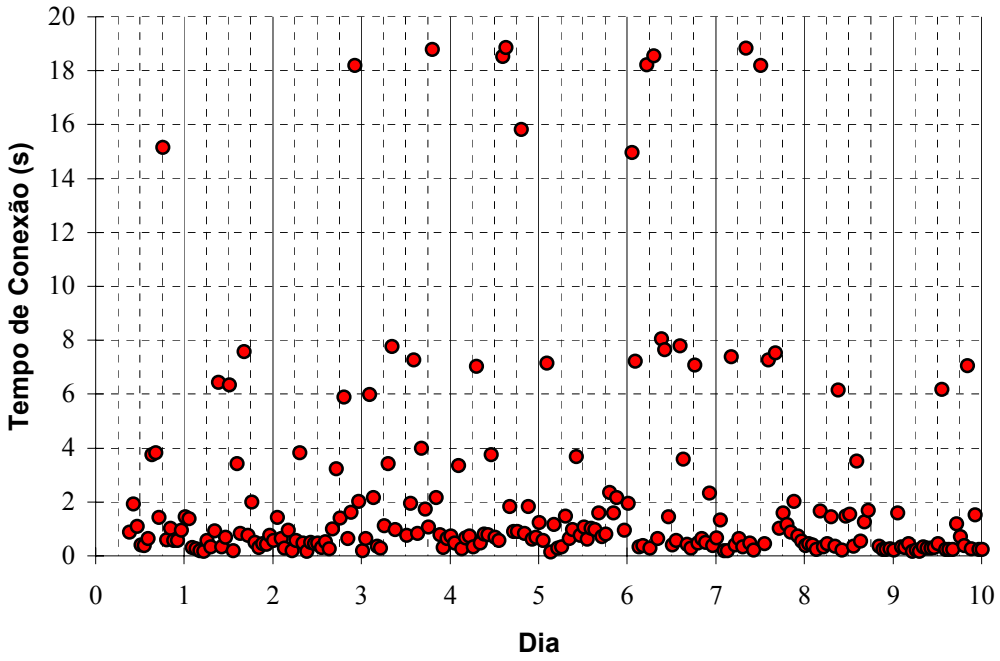


Figura 5: Tempos de Conexão de um Servidor ao Longo do Tempo.

A figura 6 apresenta a distribuição de frequências para os tempos de conexão da figura 5. Os valores de tempo de conexão situados abaixo de 2,5 s são similares ao *round-trip time* medido pelo ping, consistindo basicamente na soma do tempo de propagação de um pacote do cliente até o servidor (o pedido), o tempo de

processamento no destino, e o tempo de propagação de um pacote no sentido inverso (a resposta). Para o caso do tempo de conexão, o processamento é feito pelo servidor HTTP, enquanto que para o ping o processamento é feito a um nível mais baixo no *host*, que não está directamente relacionado com o serviço HTTP. Portanto, o tempo de conexão considera a carga no servidor mais efectivamente do que o tempo do ping.

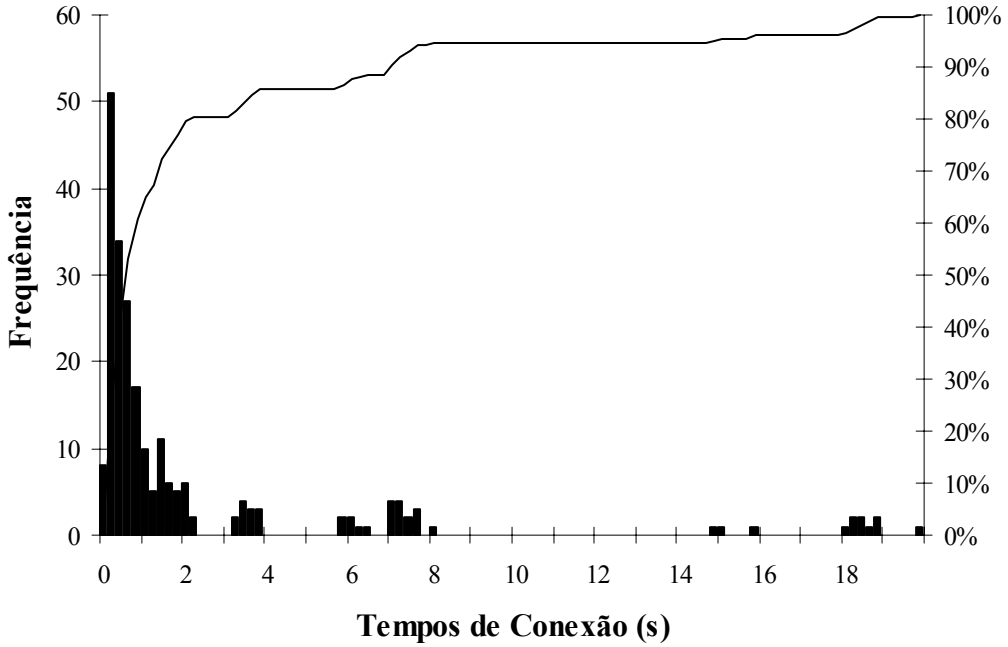


Figura 6: Distribuição de Frequências para os Tempos de Conexão de um Servidor.

Diferentemente do tempo do ping, o tempo de conexão apresenta diversas regiões com valores concentrados, acima de 2,5 s, como se pode observar na figura 6. Tal comportamento é devido a retransmissões de pacotes feitas pelo cliente ou servidor, accionadas pelo mecanismo de *timeout*. A necessidade de retransmissão ocorre em grande parte pela perda de pacotes trocados entre os dois *hosts*, descartados pelos routers no caminho devido a congestão ou outros motivos. Com o ping não há retransmissão de pacotes, portanto não há resposta caso se perca o pacote enviado por uma das partes.

A frequência de pacotes perdidos (ou de retransmissão) pode ser muito útil para a estimativa do tempo de resposta. Para servidores que apresentem *round-trip times* similares a porcentagem de pacotes perdidos na comunicação com cada um pode fazer diferença em termos de tempo de resposta.

A distribuição de frequências para os tempos de conexão de um servidor (e também para os tempos de ping) não é simétrica, sendo que a maioria das ocorrências situa-se abaixo da média. Essa assimetria é agravada, no caso dos tempos de conexão, pelas retransmissões, fazendo que a média aritmética não represente bem os tempos de conexão, principalmente quando há poucas amostras disponíveis, pois esta é muito influenciada pelos valores mais altos das amostras.

Quando se analisa os dados obtidos de uma distribuição limitada em uma direcção e ilimitada de outra, normalmente é útil aplicar uma transformação logarítmica. Se a variável aleatória $Y = \log X$ tem uma distribuição normal, então diz-se que X tem uma distribuição log-normal [46]. A transformação logarítmica reduz bastante a influência dos valores mais altos, o que interessa neste caso. Com esta transformação, a média aritmética de Y equivale à média geométrica de X.

A média geométrica de n valores é calculada multiplicando-se todos os valores e depois calculando-se a n-ésima raiz do resultado.

$$G_n = \sqrt[n]{\prod_{i=1}^{n} X_i} \tag{2}$$

Em que n é o número de amostras, G_n é a média geométrica e X_i é o valor da amostra de índice i.

Outro algoritmo foi utilizado para o cálculo da média geométrica: Primeiro faz-se a transformação logarítmica de cada amostra, depois calcula-se a média aritmética dos valores obtidos, finalmente faz-se a transformação inversa da média obtida. Trabalhar com a média aritmética facilita as operações.

$$G_n = 10^{\left(\frac{1}{n}\sum_{i=1}^{n} \log_{10} X_i\right)} \tag{3}$$

Na fórmula é utilizado o logaritmo de base 10, por ser conveniente, mas qualquer outra base pode ser utilizada.

A análise dos dados recolhidos a diversos servidores mostra que a média geométrica dos tempos de conexão fornece bons resultados.

4.4.2 MSS da Conexão

O MSS (Maximum Segment Size) utilizado nas conexões, ou seja, o tamanho máximo permitido para o campo de dados dos pacotes TCP em uma comunicação, mostrou ter influência no tempo de transferência dos dados. O MSS a ser utilizado na conexão é negociado por ambos os *hosts* durante a fase de estabelecimento de conexão. O MSS proposto por cada *host* pode ser obtido no *header* TCP dos pacotes SYN respectivos. O menor valor é utilizado pelas duas partes. Quando o MSS de uma das partes não é anunciado, considera-se o valor de 536 por *default*.

Normalmente, cada *host* possui um MSS definido, que é utilizado por todos os clientes e servidores naquele *host*. O MSS a ser utilizado em uma conexão pode ser previsto dado o MSS de cada parte. O MSS do servidor é armazenado na Tabela de QoS, enquanto o MSS do cliente pode ser fornecido por este na *query* ao servidor de QoS. Alternativamente, pode-se criar uma tabela que relacione os endereços IP dos clientes locais e respectivos MSS.

O uso de um MSS maior em uma conexão implica que mais informação pode ser transferida em cada pacote. O mecanismo de *slow-start* do TCP limita o número

de pacotes que pode ser transferido no início da conexão, e caso haja congestão em algum ponto da rede a limitação no número de pacotes pode persistir. Em tais casos, o uso de um MSS mais elevado pode reduzir significativamente o tempo de transferência, desde que não haja fragmentação dos pacotes no caminho entre os *hosts*.

Medições realizadas com diversos servidores indicaram que o uso de um MSS maior na conexão contribuiu para a redução do tempo de resposta. Assim, resolveu-se incluir na estimativa do tempo de resposta (E_{tr}) um factor (F_{MSS}) que considere a influência do MSS utilizado, tornando a estimativa proporcional a este valor.

$$E_{tr} \propto F_{MSS} \tag{4}$$

O factor de compensação do MSS (F_{MSS}) foi definido de forma que seu valor diminua com o aumento do MSS utilizado na conexão (MSS_{cx}), da seguinte forma:

$$F_{MSS} = \left(\frac{1}{MSS_{cx}} \right)^{r} \tag{5}$$

Em que o valor de r é limitado entre 0 e 1. Para $r = 1$, a estimativa do tempo de resposta seria inversamente proporcional ao valor do MSS utilizado na conexão. Quanto menor o valor de r, menor a influência do MSS na estimativa. O valor de $r = 0{,}5$ ajustou-se bem aos dados recolhidos.

4.4.3 Disponibilidade

A disponibilidade de um servidor (do ponto de vista do cliente), depende não apenas do servidor, mas em grande parte da rede. E sua estimativa é influenciada pelo protocolo utilizado nas medições, já que alguns tipos de pacotes podem ser rejeitados no caminho entre o cliente e o servidor. Sendo assim, o uso do ping para estimativa da disponibilidade de um servidor pode ser incorrecta em alguns casos, por se tratar de um protocolo diferente daquele utilizado no acesso ao recurso.

Outra desvantagem do uso do ping em relação ao tempo de conexão é que ele mede a disponibilidade do *host*, e não a disponibilidade do servidor. O *host* pode estar activo e respondendo ao ICMP *echo request*, mas o processo do servidor (HTTP) pode estar inactivo. Esta situação foi observada algumas vezes durante nossas medições.

Duas estimativas da disponibilidade de um servidor são consideradas ao se utilizar *passive probing*, tendo o tempo de conexão como métrica. A estimativa da disponibilidade a longo prazo (D_{lp}) de um servidor é obtida dividindo-se o número de repostas do servidor (N_{resp}) a pedidos de conexão feitas por clientes locais pelo número total de pedidos (N_{req}) feitos desde o início da colecta de informação. Tanto o número de pedidos quanto o número de respostas são armazenados na Tabela de QoS para cada servidor.

$$D_{lp} = \frac{N_{resp}}{N_{req}} \tag{6}$$

A estimativa da disponibilidade recente (D_{rec}) dá mais importância às últimas tentativas de conexão. Cada tentativa falhada reduz o valor da estimativa. Quando o estabelecimento de conexão é bem sucedido (há resposta por parte do servidor) a estimativa é reinicializada para o valor máximo. A estimativa da disponibilidade recente é armazenada directamente na Tabela de QoS.

$$\textit{Houve resposta}\,? \begin{cases} \textit{Sim} \;\Rightarrow\; D_{rec_t} = 1 \\ \textit{Não} \;\Rightarrow\; D_{rec_t} = D_{rec_{t-1}}\, k_d \end{cases} \tag{7}$$

D_{rec_t} é o valor da estimativa da disponibilidade recente calculado no instante t, $D_{rec_{t-1}}$ é o valor anterior da estimativa, k_d é uma constante com valor entre 0 e 1. Quanto menor o valor de κ_d, mais sensível é a estimativa à ausência de resposta por parte do servidor.

Enquanto a disponibilidade recente procura determinar se o servidor está activo no momento, a disponibilidade a longo prazo procura determinar a fiabilidade do servidor. Ambas podem ser utilizadas juntamente com a estimativa do tempo de resposta para auxiliar na escolha do servidor.

4.5 Resultados Experimentais

Com o objectivo de possibilitar a comparação entre métricas que podem ser utilizadas para a estimativa do tempo de resposta, um cliente HTTP simples foi modificado para ser capaz de recolher estatísticas sobre as conexões, dentre as quais os tempos de conexão e de resposta.

O *host* utilizado pelo cliente para um primeiro conjunto de medições foi um Soulbourne s700, com sistema operativo SunOs 4.1.3, e com MSS de 536, que está ligado à rede do Departamento de Informática da Universidade do Minho. Durante um período de 12 dias, um mesmo documento (14 KB) foi simultaneamente requisitado a um conjunto de 14 servidores HTTP distribuídos pela Internet, com 1 hora de intervalo entre os pedidos.

4.5.1 Primeiro Conjunto de Medições

Os resultados das medições realizadas com o *host* do Departamento de Informática são apresentados na tabela 1, ordenados pela média aritmética dos tempos de resposta de cada servidor, que é identificado na tabela pelo seu país. Para cada servidor, os tempos apresentados são médias de todas as medições feitas durante o período. As abreviações M.A. e M.G. foram utilizadas para a média aritmética e média geométrica, respectivamente. As falhas na transferência consistem nas falhas

em estabelecer a conexão somadas às falhas devido a timeout durante a transferência de dados (5 minutos sem recepção de informação).

Tabela 1: Resultados das Medições e Correlações para o Primeiro Conjunto de Medições.

País do servidor	T. resposta (M.A.)	Falhas na transf. (%)	Tempo de conexão		*Round-trip time*		Número de hops	MSS do servidor
			(M.G)	(M.A.)	(M.G.)	(M.A.)		
Holanda	32.60	0.4	0.53	1.12	0.41	0.57	12	1460
Alemanha	36.22	4.2	0.55	0.95	0.46	0.59	16	1460
Itália	36.88	10.6	0.66	1.38	0.50	0.63	12	1460
Espanha	42.86	5.0	0.81	3.00	0.36	0.47	14	1460
Reino Unido	51.46	3.9	1.00	3.89	0.34	0.39	16	536
França	52.23	1.5	1.09	2.84	0.63	0.80	18	1460
Noruega	62.86	13.4	1.14	3.29	0.45	0.53	14	512
Finlândia	64.35	5.0	0.98	3.85	0.35	0.43	11	512
U.S.A. (4)	70.00	2.1	1.56	3.72	0.69	0.80	20	1460
U.S.A. (2)	94.37	6.8	1.73	4.99	0.63	0.70	19	1460
U.S.A. (3)	101.45	15.3	1.74	5.16	0.65	0.70	20	1460
Áustria	101.64	10.0	1.42	4.19	0.58	0.65	14	536
U.S.A. (1)	110.98	12.4	1.69	5.17	0.64	0.69	19	1460
Rep. Checa	138.02	11.4	2.46	7.17	0.78	0.86	14	512
Correlação com média dos t. resposta			0.95	0.93	0.75	0.56	0.34	

Para calcular as médias dos *round-trip times* utilizou-se os tempos de conexão inferiores a 2.5 s. Os tempos superiores a este valor foram descartados por serem obtidos através de retransmissão de pacotes. Desta forma, obtém-se um resultado similar ao que seria obtido utilizando-se o ping.

A medição do número de *hops* entre o cliente e cada servidor foi feita utilizando-se o traceroute.

O MSS do cliente (536) é utilizado nas conexões com todos os servidores da tabela 1, excepto aqueles que possuem MSS de 512. Nestes casos o valor de 512 é utilizado por ser menor. De qualquer forma, os dois valores são muito próximos, portanto a influência do MSS para efeito de comparação entre os servidores é mínimo, não sendo considerada aqui.

4.5.1.1 Resultados Globais

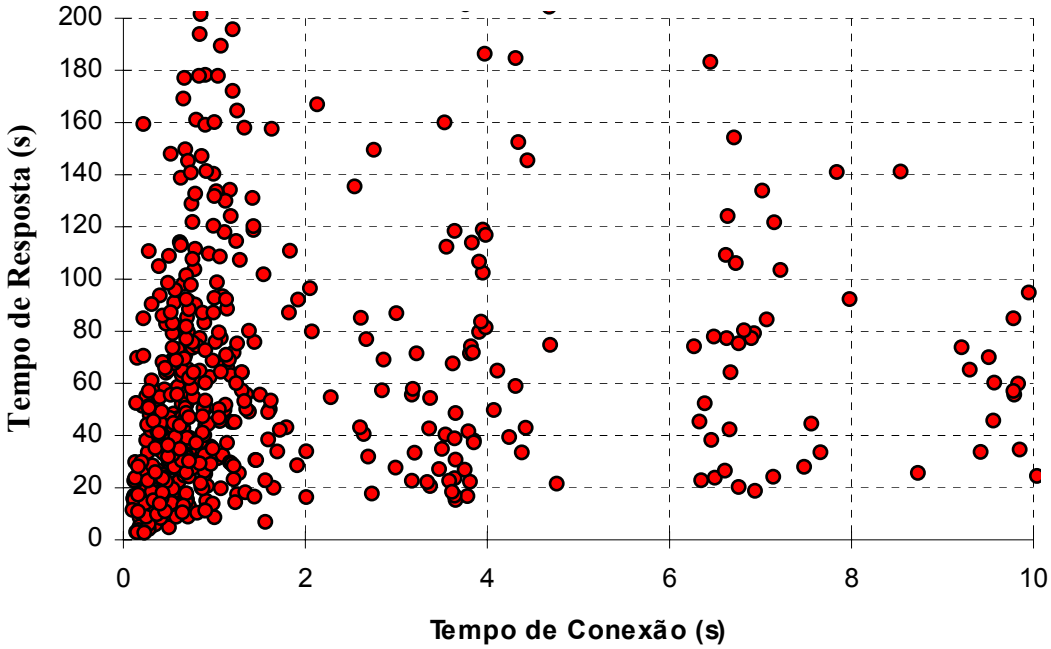


Figura 7: Gráfico da Relação entre os Tempos de Resposta e Respectivos Tempos de Conexão, Incluindo as Medições Feitas a Todos os Servidores.

A figura 7 apresenta a relação entre os tempo de resposta e os respectivos tempos de conexão, reunindo os valores obtidos para todos os servidores (alguns valores excederam os limites do gráfico). Nota-se, para os tempos de conexão, as concentrações de valores originadas pela retransmissão de pacotes. Os pontos no gráfico estão muito dispersos, consequentemente, a correlação entre os tempos de resposta e os respectivos tempos de conexão é apenas 25 %. Esse resultado indica que uma única medição do tempo de conexão não é uma boa estimativa nem mesmo do tempo de resposta associado.

Por outro lado, as médias dos tempos de resposta e tempos de conexão de cada servidor apresentam uma correlação muito maior (95 % para a média geométrica e 93 % para a média aritmética dos tempos de conexão). A figura 8 apresenta a relação entre os tempos de conexões médios e a média geométrica dos tempos de conexão respectivos. Neste caso, nota-se uma recta muito melhor definida do que no caso da figura 7.

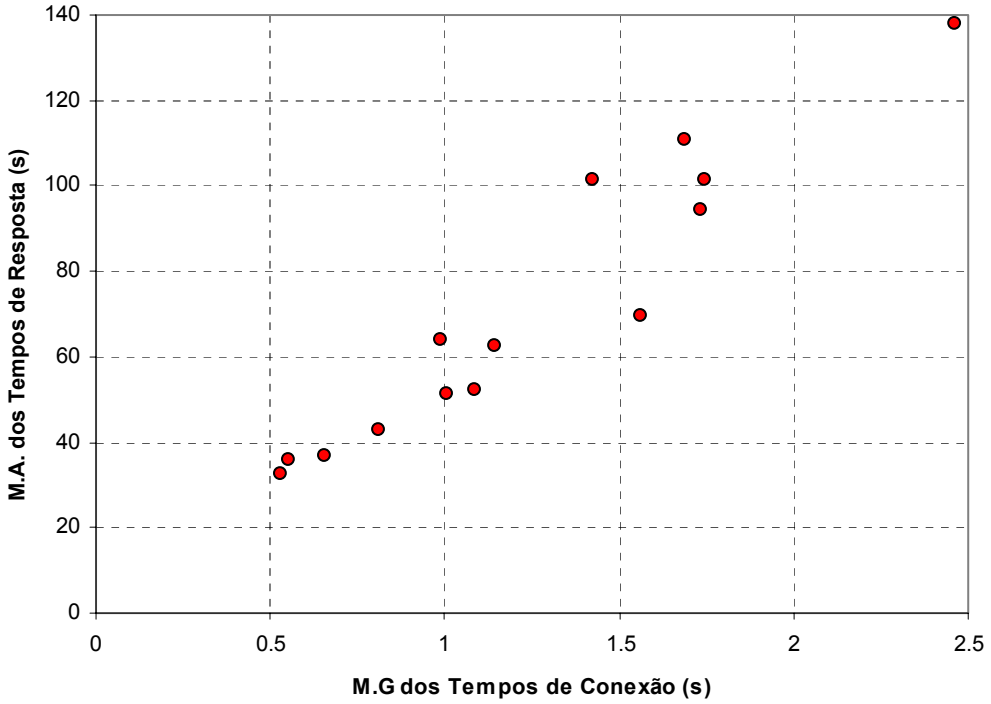


Figura 8: Gráfico da Relação entre a Média Aritmética dos Tempos de Resposta de Cada Servidor e a Média Geométrica dos Tempos de Conexão Respectivos.

Essas correlações também são superiores às obtidas com as médias dos *round-trip times* (75 % para M.G. e 56 % para M.A.). Uma consequência disso é que a ordem de selecção utilizando-se os tempos de conexão médios é bem mais próxima da ordem dos tempos de resposta médios do que a ordem de selecção baseada na média dos *round-trip times*. No caso apresentado na tabela 1, a sequência das médias aritméticas dos tempos de conexão é igual à sequência das médias aritméticas dos tempos de resposta para os quatro primeiros servidores. Por outro lado, o primeiro servidor escolhido com base em ambas as médias dos *round-trip times* corresponde ao quinto servidor com menor tempo de resposta médio.

A correlação entre os tempos de resposta médios e o número de *hops* é de apenas 34 %. Alem disso, o servidor mais próximo em termos de número de *hops* é apenas o oitavo com menor tempo de resposta médio. Estes resultados confirmam que o número de *hops* é uma má estimativa do tempo de resposta.

O tempo de resposta médio óptimo (seleccionando-se o menor tempo de resposta em cada intervalo de medições) foi 21,2 s para este conjunto de medições. Este resultado é apenas 35 % inferior ao valor de 32,6 s obtido para o servidor da Holanda. Por outro lado, o tempo de resposta médio péssimo foi 147,6 s (4,3 vezes maior que o tempo do servidor holandês), enquanto que a porcentagem de falhas na transferência foi 50 % (125 vezes maior).

O tempo de resposta médio para todos os servidores é 71,2 s, e a média de falhas na transferência é 7,3 %. Estes seriam os valores obtidos em média se o critério de selecção aleatória do servidor fosse utilizado. O uso do servidor da Holanda durante o mesmo período proporciona uma redução em 54 % do tempo de

resposta médio e de 95 % nas falhas em relação ao uso da selecção aleatória, alcançando o objectivo definido anteriormente nesta tese.

Para a comparação entre os resultados obtidos com a média dos tempos de conexão e os resultados que seriam obtidos caso se utilizasse a estimativa instantânea, seleccionou-se, a cada período, o tempo de resposta do servidor com menor tempo de conexão. A média dos tempos de resposta utilizando-se este método de selecção foi 38,0 s, superior ao valor de 32,6 s obtido para o servidor da Holanda e ao valor dos dois servidores que lhe seguem. Isso demonstra que o *probing* de todos os servidores imediatamente antes da selecção, e a escolha do servidor com melhor estimativa naquele instante pode produzir resultados piores do que simplesmente optar-se pelo mesmo servidor durante um longo período. Os resultados ainda poderiam ser piores com o uso do ping, já que a estimativa feita aqui utilizou os tempos de conexão, que fazem parte do tempo de resposta e consideram a carga no servidor.

4.5.1.2 Divisão das Medições em Duas Partes

Na tabela 1, as médias dos tempos de conexão foram comparadas com as médias dos tempos de resposta respectivos, com bons resultados. Porém, o que se pretende é utilizar a media dos tempos de conexão para prever o comportamento dos tempos de resposta em conexões futuras. Para avaliar a eficiência do método neste caso, dividiu-se o conjunto de medições em duas partes de 6 dias cada. A tabela 2 apresenta os resultados para os primeiros 6 dias, e a tabela 3 apresenta os resultados para os 6 dias restantes, de forma idêntica à apresentada na tabela 1.

Tabela 2: Resultados das Medições e Correlações para o Primeiro Conjunto de Medições (Primeira Metade).

País do servidor	T. resposta (M.A.)	Falhas na transf. (%)	Tempo de conexão		*Round-trip time*		Número de hops
			(M.G)	(M.A.)	(M.G.)	(M.A.)	
Holanda	29.00	0.7	0.58	1.22	0.43	0.63	12
Alemanha	34.17	5.0	0.56	0.81	0.50	0.62	16
Itália	42.34	5.0	0.69	1.22	0.53	0.67	12
Espanha	43.35	0.7	0.84	3.06	0.37	0.48	14
Reino Unido	55.55	0.0	1.18	4.67	0.35	0.41	16
França	60.01	0.0	1.21	3.02	0.69	0.86	18
Finlândia	75.12	2.9	1.29	4.85	0.40	0.48	11
Noruega	76.34	10.8	1.35	3.75	0.49	0.58	14
U.S.A. (4)	78.16	1.4	1.61	3.83	0.69	0.77	20
U.S.A. (2)	107.85	5.8	1.87	5.55	0.66	0.75	19
Áustria	116.56	9.4	1.58	4.29	0.57	0.66	14
U.S.A. (1)	117.32	14.4	1.76	5.98	0.64	0.70	19
U.S.A. (3)	118.98	15.9	2.06	6.34	0.66	0.72	20
Rep. Checa	139.55	11.6	2.64	7.75	0.80	0.89	14
Correlação com média dos t. resposta			0.95	0.90	0.69	0.50	0.39

Tabela 3: Resultados das Medições e Correlações para o Primeiro Conjunto de Medições (Segunda Metade).

País do servidor	T. resposta (M.A.)	Falhas na transf. (%)	Tempo de conexão		*Round-trip time*		Número de hops
			(M.G)	(M.A.)	(M.G.)	(M.A.)	
Itália	30.97	16.6	0.63	1.55	0.46	0.59	12
Holanda	35.82	0.0	0.49	1.02	0.39	0.52	12
Alemanha	37.96	3.5	0.55	1.08	0.43	0.57	16
Espanha	42.09	9.0	0.78	2.92	0.35	0.45	14
França	44.42	2.8	0.97	2.64	0.58	0.74	18
Reino Unido	48.76	7.5	0.90	3.42	0.33	0.38	16
Noruega	49.76	15.9	0.96	2.83	0.41	0.50	14
Finlândia	53.35	6.9	0.75	2.82	0.32	0.38	11
U.S.A. (4)	61.72	2.8	1.51	3.60	0.70	0.82	20
U.S.A. (2)	80.71	7.6	1.60	4.42	0.60	0.66	19
U.S.A. (3)	84.51	14.5	1.48	4.02	0.64	0.69	20
Áustria	86.72	10.3	1.30	4.11	0.58	0.64	14
U.S.A. (1)	105.67	10.3	1.62	4.54	0.65	0.69	19
Rep. Checa	135.79	11.0	2.33	6.62	0.77	0.83	14
Correlação com média dos t. resposta			0.94	0.93	0.77	0.60	0.32

Pode-se observar resultados bastantes semelhantes entre tabelas. A correlação entre os tempos de resposta médios para as duas partes é 94 %, indicando a estabilidade dos tempos de resposta dos servidores ao longo do tempo.

A correlação entre a média geométrica dos tempos de conexão da primeira parte e a média dos tempos de resposta da segunda é 92 % (não indicada na tabela), ainda bastante elevada, indicando que a validade das estimativas mantém-se neste caso. O servidor escolhido passa a ser o da Alemanha (38,0 s), por pouca diferença em relação ao da Holanda. O servidor da Itália passou a apresentar o menor tempo de resposta médio, entretanto, a porcentagem de falhas na transferência (não consideradas na média) foi expressiva no período. A escolha do servidor da Alemanha ainda é boa, pois a média dos tempos de resposta de todos os servidores é 64,2 s (69 % maior). Os resultados obtidos com a estimativa instantânea são piores também (40,8 s).

Para avaliação de resultados, pode-se também fazer a comparação no sentido inverso. Assim, a correlação entre a média geométrica dos tempos de conexão da segunda parte e a média dos tempos de resposta da primeira é de 91 %, da mesma forma elevada. O servidor escolhido passa a ser novamente o da Holanda, com 29,0 s (o primeiro do *ranking*). A média do tempos de resposta globais é 78,2 s (170 % mais elevada), e a média dos tempos de resposta dos servidores escolhidos através da estimativa instantânea é também maior (35,0 s).

4.5.1.3 Conclusões

Esses resultados demonstram que alguns servidores apresentam consistentemente melhores tempos de resposta do que outros. Assim, mesmo se um

servidor for utilizado por um longo período de tempo, se a escolha for correcta, a redução do tempo de resposta pode ser considerável.

Isso não significa que não se possa (ou deva) mudar de servidor com o uso do *passive probing*. Tal restrição é típica de um método de selecção estático, mas o método proposto é dinâmico, de forma que se o servidor em primeiro lugar no *ranking* (dentre um conjunto de servidores) começar a se comportar mal, sua posição no *ranking* mudará, e o servidor seguinte passará a ser o escolhido.

Outra conclusão que pode ser retirada destes resultados é que os *round-trip times* produzem estimativas menos correctas que os tempos de conexão, principalmente se os pacotes perdidos e retransmissões não são considerados.

Os resultados também demonstram que, apesar de não necessitar de inserir tráfego na rede para calcular as estimativas, o método de selecção baseado no *passive probing* pode produzir melhores resultados do que alguns métodos baseados na estimativa instantânea, nomeadamente em casos em que a estimativa baseia-se em uma única medição do *round-trip time*.

4.5.2 Segundo Conjunto de Medições

Medições semelhantes às efectuadas anteriormente foram feitas com o cliente em outro *host*. O *host* utilizado para este segundo conjunto de medições foi um SunServer 10 com SunOs 5.3 e MSS de 1460, ligado à rede do Grupo de Comunicações por Computador da Universidade do Minho.

O tempo de resposta médio óptimo para este conjunto de medições foi 15,3 s, enquanto que o péssimo foi 199,7 s, mais 44,9 % de falhas na transferência. O tempo de resposta médio de todos os servidores foi 73,2 s. O servidor escolhido pela média geométrica dos tempos de conexão foi o da Alemanha, que apresentou tempo de resposta médio de 28,8 s, logo a seguir ao servidor da Holanda, o primeiro da lista, com 26,6 s. A redução no tempo de resposta ao se utilizar o servidor da Alemanha, em relação à média de todos os servidores, é de 61 %. Mais uma vez, a estimativa instantânea apresentou pior resultado, obtendo um tempo de resposta médio de 41,2 s com suas escolhas.

Neste caso, o MSS utilizado em cada conexão foi o próprio MSS do servidor, mostrado na tabela 1, já que o MSS do cliente é igual ou superior ao MSS de todos os servidores medidos. Os resultados da correlação entre os tempos de resposta médios e os tempos de conexão médios, *round-trip times* médios e número de *hops* são apresentados na primeira linha da tabela 4. Os cálculos são similares aos apresentados na tabela 1, a correlação encontrada, porém, é bastante inferior em todos os casos. O número de *hops* chega a apresentar uma correlação negativa.

Tabela 4: Segundo Conjunto de Medições - Correlação entre as Médias dos Tempos de Resposta e Respectivas Estimativas, Antes e Depois de Incluir o MSS da Conexão.

Correlação com a Média dos Tempos de Resposta	Tempo de conexão		*Round-trip time*		Número de hops
	(M.G)	(M.A)	(M.G)	(M.A)	
Considerando apenas medições	0.50	0.65	0.23	0.06	-0.10
Considerando também o MSS da conexão	0.79	0.89	0.77	0.73	0.64

O exame dos dados recolhidos mostrou que a ordem dos tempos de resposta médios também se alteraram, sendo que os servidores com MSS da 1460 melhoraram a sua posição no *ranking* em relação as servidores com MSS de 536 e 512. Houve também uma redução, em média, dos tempos de resposta para um mesmo servidor, quando era utilizado um MSS de 1460 na conexão, em relação ao uso do MSS de 536.

Para cada servidor, aplicou-se o factor de compensação do MSS (F_{MSS}), definido na equação 5 (com $r = 0{,}5$) às estimativas feitas com os tempos de conexão, *round-trip times* e número de *hops*, obtendo-se novas estimativas. A correlação entre os tempos de resposta médios e essas novas estimativas é apresentada na segunda linha da tabela 4. Pode-se observar uma melhoria significativa da correlação em todos os casos quando se considera a influência do MSS.

Também utilizou-se o F_{MSS} para corrigir o valor da estimativa instantânea com bons resultados, conseguindo-se reduzir o tempo de resposta médio de 41,2 s para 33,0 s.

4.6 Outras Considerações sobre a Selecção

4.6.1 Gestão dos Nomes de Domínio

Em [35] e [36], os autores assumem que uma lista de endereços IP deve ser obtida para proceder-se ao processo de selecção. Entretanto, na maior parte dos casos o cliente tem acesso primeiramente aos nomes dos servidores. Para conhecer os endereços IP, o cliente terá que contactar o serviço de DNS para cada servidor, causando com isso dois inconvenientes:

- Desperdício de recursos da rede, pois diversas *queries* serão feitas ao DNS, uma para cada servidor na lista, mas apenas um servidor será seleccionado.

- Grande demora, pois o cliente necessita esperar pelas respostas a todas as queries ao DNS para obter a lista completa de endereços IP e poder submetê-la ao processo de selecção.

Nós consideramos que, para que o processo de selecção do servidor seja eficiente, o serviço de QoS deve ser capaz de trabalhar não só com endereços IP dos servidores, mas também com os nomes dos servidores, indiferentemente. Entretanto, isso deve ser possível sem que haja a necessidade de se fazer *queries* ao DNS, ou os inconvenientes persistiriam. A solução consiste em armazenar na Tabela de QoS

tanto o nome quanto o endereço IP dos servidores, recolhidos usando-se *passive probing*, à medida em que são feitas novas conexões.

Entretanto, os pacotes do HTTP apenas contém o endereço IP do servidor, não o seu nome de domínio. O tcpdump possui uma opção para converter endereços IP em nomes, mas o seu uso apresenta problemas:

- O tcpdump retorna o nome canónico do *host,* mas normalmente o nome pelo qual o servidor é conhecido é um alias.
- Para obter o nome do servidor, o tcpdump utiliza os serviços do DNS.
- Por vezes a resolução do nome do *host* falha.
- A tentativa de obter os nomes dos *hosts* provoca um aumento do número de pacotes perdidos pelo tcpdump.

A solução para esses problemas é obtida monitorando-se também o tráfego DNS com o tcpdump. Antes de contactarem um servidor, os clientes habitualmente fornecem o seu nome ao DNS para obterem o endereço IP respectivo. Comparando-se os endereços IP contidos tanto nos pacotes HTTP quanto nos pacotes de resposta do DNS consegue-se determinar o nome do servidor que foi utilizado pelo cliente. Esse nome passa a ficar registado na Tabela de QoS, associado ao endereço IP do servidor.

Um benefício adicional que o procedimento de se armazenar os nomes e endereços IP dos servidores pode trazer é que, como o serviço de QoS pode fornecer o endereço IP do servidor a partir do seu nome, a *query* normalmente feita pelo cliente ao servidor de DNS pode ser evitada.

4.6.2 Obtenção dos Dados da Tabela de QoS

Caso a Tabela de QoS seja mantida junto a um *proxy*, a informação fica prontamente disponível aos clientes por intermédio do *proxy*. Este pode obter a lista de servidores, consultar a Tabela de QoS, contactar o servidor escolhido, receber o recurso desejado e repassá-lo ao cliente; que não necessita de intervir no processo. Desta forma, mesmo clientes simples podem beneficiar-se do processo de selecção.

Existem muitos clientes que não utilizam os serviços de um *proxy*. Para possibilitar o acesso à informação contida na Tabela de QoS a todos os clientes pode-se criar um serviço independente, acessível utilizando-se um protocolo apropriado, como proposto em [35].

O serviço, de nome SONAR, foi criado pelos autores com o objectivo de fornecer estimativas de proximidade entre hosts, sendo implementado sobre o UDP. O servidor SONAR atende a pedidos na porta 572. Tanto os pedidos (*queries*) como as respostas utilizam apenas um datagrama UDP cada. O formato da *query* do SONAR inclui um identificador para associar uma resposta com a *query* respectiva; este identificador obviamente é incluído também na resposta. Uma configuração

semelhante a essa poderia ser utilizada para um servidor de QoS, que atenderia a pedidos de informação da Tabela de QoS.

A *query* contém um campo de *timeout*, no qual o cliente deve informar o número máximo de segundos que o servidor SONAR deve esperar antes de enviar uma resposta, mesmo que esteja incompleta. O motivo da existência desse campo é que o método utilizado para o cálculo das estimativas utiliza *active probing*, não havendo garantias quanto ao tempo necessário à obtenção das estimativas. Esse campo é totalmente dispensável no nosso caso, pois as estimativas já estão prontas quando a *query* é recebida.

A *query* do SONAR contém ainda uma lista de endereços IP para os quais se deseja a estimativa de proximidade. No nosso caso admite-se também a utilização dos nomes dos servidores.

A resposta do SONAR é composta basicamente por pares <endereço IP, estimativa>. Como a Tabela de QoS tem uma variedade maior de informação a fornecer, o formato utilizado pode ser um pouco mais complexo, sem deixar de ser simples.

Utilizando-se um *host* local para o fornecimento do serviço de QoS, o custo de se fazer uma *query* à tabela de QoS é apenas a troca de um par de pacotes entre dois *hosts* locais (com uma duração típica de poucos milisegundos). Os benefícios podem ser uma considerável redução no tempo total necessário à obtenção de um ou mais recursos, e um encaminhamento mais eficiente do tráfego de centenas de pacotes.

5 Implementação

A figura 9 apresenta a estrutura da rede de comunicações da Universidade do Minho [41], com destaque para as redes utilizadas nesta implementação. A recolha do tráfego é feita na rede que faz a ligação do *router* da Universidade do Minho com o router BRAGA-ROUTER-6.rccn.net, que faz o acesso com o exterior, conforme indicado na figura 9. Todo o tráfego originado na Universidade do Minho com destino externo, e vice-versa, passa por esse ponto.

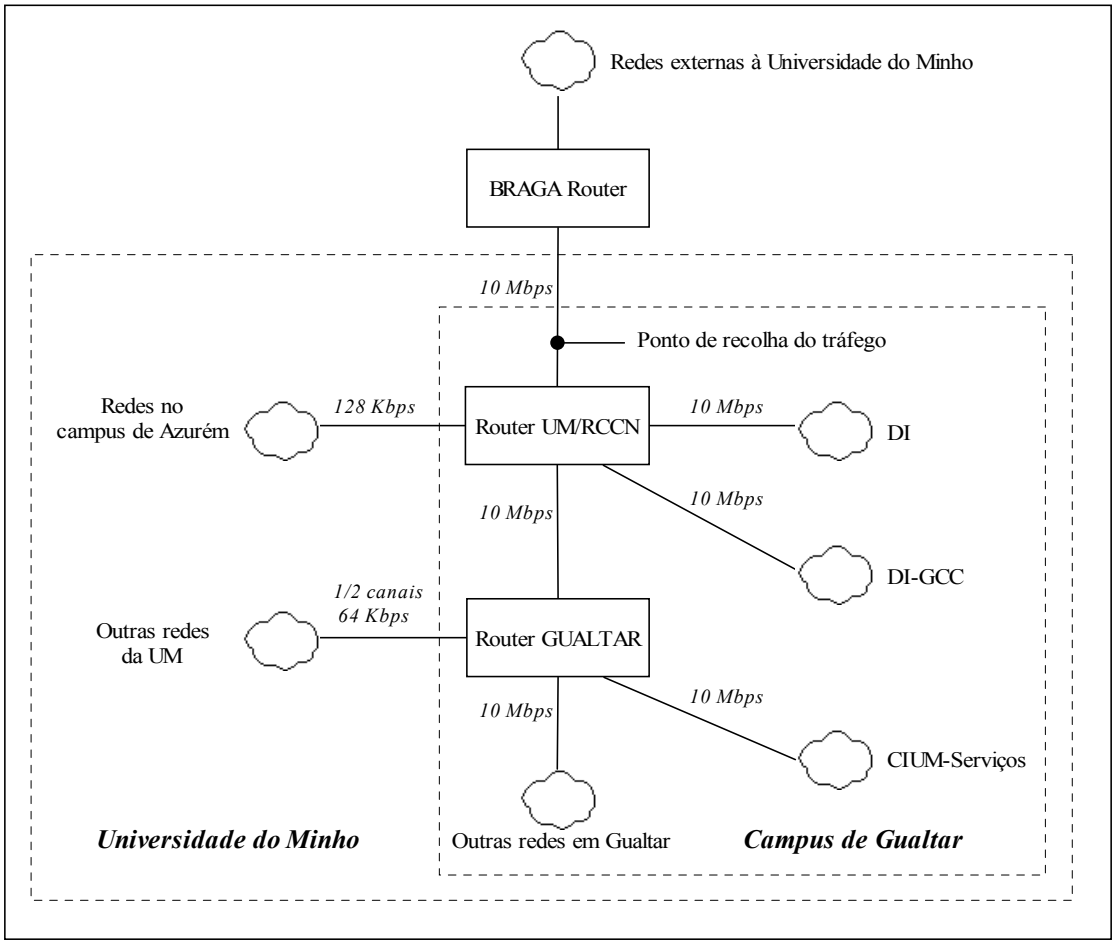


Figura 9: Estrutura da Rede de Comunicações da Universidade do Minho.

Na época de início da recolha havia um *host* conectado a essa rede disponível e com o tcpdump instalado, sendo naturalmente escolhido para executar a recolha do tráfego. O *host*, de nome viriato, é um SunSparc, com SunOS 4.1.1.

O campus de Gualtar da Universidade do Minho apresenta diversas redes Ethenet com débito de 10 Mbps conectadas entre si. Dentre essas redes, foram escolhidas três para a recolha do tráfego gerado por seus clientes: a rede 193.136.20, do DI (Departamento de Informática), a rede 193.136.9, do GCC (Grupo de Comunicações por Computador), e a rede 193.136.16, uma das redes do CIUM (Centro de Informática da Universidade do Minho). Existem outras redes localizadas no campus de Gualtar que também podem ser incluídas na recolha de tráfego (CIUM-Alunos, SDUM, etc.), entretanto, as redes escolhidas são suficientes para testar o funcionamento da implementação.

O tráfego iniciado por clientes de outras redes que também fazem parte da Universidade do Minho, mas são externas ao campus de Gualtar, não é utilizado, como é o caso das redes do campus de Azurém, do Largo do Paço e da Gulbenkian, Da mesma forma, o tráfego iniciado por clientes externos à Universidade é rejeitado.

5.1 Obtenção e Armazenamento da Informação

Existem três etapas necessárias à produção da informação de QoS, que podem ser resumidas nas seguintes tarefas:

- Recolha do tráfego.
- Identificação das conexões.
- Actualização da Tabela de QoS.

Cada tarefa é executada por um programa diferente. Os programas funcionam simultaneamente, em *pipeline*, durante 24 horas por dia, fazendo com que a informação seja continuamente actualizada.

5.1.1 Recolha do Tráfego

A tarefa de recolher o tráfego cabe ao tcpdump, sendo que o filtro utilizado deixa passar apenas os pacotes do HTTP e do DNS. Para descobrir o nome de domínio utilizado em uma conexão feita por um cliente local interessam apenas os pacotes do DNS destinados a redes locais e que sejam respostas a queries. Sendo assim, a linha de comando utilizada para o tcpdump é a seguinte:

tcpdump -n -x -s174 (port 80 and 'tcp[13] & 2 != 0') or (port 53 and dst net 193.136 and 'udp[10:2] & 0xF80F != 0x8000')

A opção *-n* é utilizada para evitar que o tcpdump tente fazer a conversão dos endereços IP em nomes. A opção *-s174* é usada para garantir que o tcpdump capture uma quantidade suficiente de bytes de modo a possibilitar a obtenção da informação desejada dos pacotes do DNS, e a opção *-x* é usada para que o tcpdump apresente os bytes capturados (apesar de o tcpdump examinar os pacotes de DNS e fornecer alguma informação sobre o seu conteúdo, esta informação geralmente não é suficiente para se obter os nomes e endereços IP associados às *queries*).

Uma breve explicação da composição da expressão booleana que define o filtro é apresentada a seguir:

- *port 80* - É a porta utilizada pelo HTTP, por *default*.
- *tcp[13] & 2 != 0* - Selecciona os pacotes SYN do TCP, os únicos necessários à medição do tempo de conexão.
- *port 53* - É a porta utilizada pelo DNS.
- *dst net 193.136* - Selecciona apenas pacotes que tenham como destino uma das redes com prefixo 193.136, caso das redes locais da Universidade do Minho.
- *udp[10:2] & 0xF80F != 0x8000* - Selecciona apenas pacotes de resposta do DNS que não contém erros. Esses pacotes contém tanto a questão quanto a sua resposta, sendo suficientes para se obter o nome do servidor e o respectivo endereço IP.

5.1.2 Identificação das Conexões

A saída do tcpdump é enviada através de um *pipe* para o programa getcx, encarregado da identificação das conexões. Este programa, feito em C [42], possui também um filtro para que sejam consideradas apenas conexões iniciadas por clientes de redes locais ao campus de Gualtar da Universidade do Minho, rejeitando-se o tráfego originado por clientes externos. O tráfego que tem origem em outras redes internas à Universidade do Minho, mas que estão afastadas do campus de Gualtar (e, consequentemente, do ponto de recolha do tráfego) também é rejeitado, pois a latência adicional no trajecto dos pacotes poderia influir nas medições. Apesar dos clientes dessas redes não contribuírem para a Tabela de QoS, ainda poderão utiliza-la, já que o caminho até esses clientes é comum a todos os servidores externos, portanto, o ranking dos servidores tende a não sofrer alterações para esses clientes.

A selecção dos clientes a serem utilizados na recolha de tráfego é feita com base nos endereços IP das redes. Como tal, poderia ter sido implementada, alternativamente, no filtro do tcpdump.

O programa getcx produz em sua saída informação sobre cada conexão que termina. Os principais campos desta informação são os seguintes:

- O endereço IP do servidor.

- O nome de domínio do servidor (caso tenha sido capturado naquele momento).
- O tempo de conexão (ou um identificador de que a tentativa de conexão falhou).
- O MSS do servidor

Outros campos disponíveis são:

- O momento de início ou fim da conexão (identificando o dia e hora exacta, com a precisão fornecida pelo *timestamp* do tcpdump).
- O endereço IP do cliente
- O MSS do cliente (com essa informação pode-se criar uma tabela relacionando o endereço IP e o MSS dos clientes locais).
- A porta do servidor (útil caso se deseje discriminar as conexões por protocolo, mas sem importância para esta implementação, que monitora as conexões de um único protocolo).

5.1.3 Actualização da Tabela de QoS

A saída do programa getcx também é enviada através de um *pipe* para a etapa seguinte, o programa regista, também feito em C. Este é encarregado de receber a informação referente a cada conexão e actualizar a Tabela de QoS, que é mantida em disco.

A Tabela de QoS ocupa muito pouco espaço em disco. Na configuração actual, definida para 50.000 endereços IP e igual número de nomes de domínio, ocupa cerca de 8 Mb. A informação relativa a cada servidor ocupa, portanto, cerca de 160 bytes.

A configuração da linha de comando englobando as três etapas é a seguinte:

tcpdump <opções> <filtro> | getcx | regista

5.1.3.1 Estrutura da Tabela de QoS

A base de dados contendo informação de qualidade de serviço foi implementada utilizando-se tabelas HASH, para permitir a rápida obtenção dos dados. A tabela principal é indexada pelo endereço IP do servidor, e contém os seguintes campos:

- O endereço IP do servidor.
- A estimativa do tempo de resposta baseada na média dos tempos de conexão (E_{cx}).

- A estimativa da disponibilidade recente do servidor (D_{rec}).
- O MSS do servidor (MSS_s).
- O número de pedidos de conexões feitas por clientes locais desde o início das medições (N_{req}).
- O número respostas a pedidos de conexões por parte do servidor durante o mesmo período (N_{resp}).
- Um *timestamp* relativo ao tempo em que foi feita a última conexão.
- Um ponteiro para uma lista de nomes de domínio associados a este servidor.

Outra tabela HASH indexada pelos nomes de domínio dos servidores contém os seguintes campos:

- Um nome de domínio.
- Um ponteiro para o registo associado ao servidor na tabela principal.
- Um *timestamp* relativo à última vez que o nome foi referenciado em uma conexão.

5.1.3.2 Gestão de Nomes e Endereços IP

Na Internet, um endereço IP frequentemente está associado a diversos nomes de domínio (*aliases*). Existem também casos, mais raros, em que um nome de domínio está associado a mais de um endereço IP.

Tendo isso em conta, a informação de QoS é associada directamente ao endereço IP na Tabela de QoS, mas a gestão dos nomes de domínio é feita separadamente, sendo a sua relação com os endereços IP definida por apontadores. A quantidade de nomes armazenados na Tabela de QoS costuma ser ligeiramente maior do que a quantidade de endereços IP, o que é normal, devido aos *aliases*.

A informação de QoS é indexada tanto pelo nome quando pelo endereço IP. Quando a procura é feita através do endereço IP, vai-se directo ao registo que contém a informação na tabela de dados. Quando a procura é feita através do nome, vai-se primeiramente à posição na tabela de nomes que contém o apontador para o respectivo registo na tabela de dados. Esta implementação é sensível a alterações na informação fornecida pelo DNS. Caso um nome de domínio passe a referenciar outro endereço IP no DNS, o valor do apontador na Tabela de QoS é actualizado logo que o novo endereço é utilizado.

5.1.3.3 Média dos Tempos de Conexão

Para o cálculo da média dos tempos de conexão de um servidor, é preferível utilizar-se uma média móvel em vez da média simples, pois as amostras muito antigas não tem grande valor para o cálculo da estimativa. A média móvel á calculada com a fórmula.

$$MM_t = \frac{X_t + X_{t-1} + \ldots + X_{t-N+1}}{N} = \frac{1}{N} \sum_{i=t-N+1}^{t} X_i \tag{8}$$

Em que MM_t é o valor da média móvel no tempo t, X_i é o valor da amostra no tempo i, N é o número de amostras incluídas na média. Quanto maior N, mais suave é a média ao longo do tempo.

O inconveniente do uso desta média na implementação é que é necessário manter armazenadas as N últimas amostras de cada servidor para o cálculo da média. Uma alternativa melhor consiste no uso do *single exponential smoothing* [43] para o cálculo da média, da seguinte forma:

$$S_t = \alpha X_t + (1-\alpha) S_{t-1} \tag{9}$$

Em que α é uma constante com valor entre 0 e 1, X_t é o valor da amostra no tempo t, S_t é o valor da média no tempo t, calculada com o *single exponential smoothing*, e S_{t-1} é o valor anterior da média.

Uma das características desta média é que todas as amostras são consideradas, porém, quanto mais antiga for a amostra, menor é o seu peso na média. O peso das amostras decresce exponencialmente, sendo que quanto menor o valor de α, mais suave é a exponencial e, consequentemente, a média ao longo do tempo.

A principal vantagem do uso do *single exponential smoothing* é que para o cálculo do novo valor da média basta ter armazenado o valor da última média (o valor da nova amostra é utilizado no cálculo, mas não precisa ser armazenado).

O cálculo da média utilizando-se o *single exponential smoothing* necessita de uma valor inicial S_0. Se este valor for mal escolhido, pode demorar muito tempo até a média alcançar o valor correcto. Para minimizar este problema, optou-se por utilizar a média simples para as primeiras N amostras, passando-se depois a utilizar o *single exponential smoothing*. Com isso, consegue-se uma precisão maior na estimativa quando o número de amostras disponível é pequeno.

Pode-se calcular a média simples utilizando-se a mesma fórmula do *single exponential smoothing*, mas em vez de se utilizar um coeficiente α constante, faz-se:

$$\begin{cases} \alpha = \dfrac{1}{n}, & se \;\; n \le N \\ \alpha = \dfrac{1}{N}, & se \;\; n > N \end{cases} \tag{10}$$

Em que n é o número de amostras recolhidas até o momento. Desta forma, consegue-se calcular a média simples de n amostras sem precisar armazená-las. No

momento em que $n = N$, a transição para o *single exponential smoothing* faz-se de modo suave.

A estimativa dos tempos de resposta é calculada com base na média geométrica dos tempos de conexão, não na média aritmética. Por isso, o cálculo que é feito para cada nova amostra é o seguinte:

$$LGS_t = \alpha \log_{10} X_t + (1-\alpha) LGS_{t-1} \quad (11)$$

Em que LGS_t é o valor do logaritmo da média geométrica no tempo t, LGS_{t-1}, é o valor anterior do logaritmo da média. O que é realmente armazenado é o logaritmo da média geométrica. O valor da média é calculado quando a consulta à Tabela de QoS é feita, da seguinte forma:

$$GS_t = 10^{LGS_t} \quad (12)$$

Em que GS_t é a média geométrica dos tempos de conexão, calculada com o uso do *single exponential smoothing*.

5.2 Problemas Enfrentados

5.2.1 Perda de Pacotes

Durante a implementação foram enfrentados diversos problemas relacionados com a perda de pacotes pelo tcpdump. Depois de muitas tentativas, as soluções foram encontradas.

A utilização do viriato teve que ser compartilhada, devido à existência de um trabalho referente à outra tese que também necessitava de recolher o tráfego do HTTP. Nesse caso, o tráfego recolhido era armazenado em disco para processamento posterior.

A configuração inicial utilizada era a seguinte:

a) Para esta tese, uma instância do tcpdump sendo executada no viriato, juntamente com os programas getcx e regista.

b) Para a outra tese, uma outra instância do tcpdump (com filtro diferente), cuja saída era redireccionada para um ficheiro em outro *host* (hermes, um DEC-Alpha com OSF/1 v2.0) via NFS (pois o viriato não possuía espaço em disco suficiente).

Com esta configuração, a perda de pacotes em ambas as instâncias do tcpdump era bastante elevada (por volta de 15 %). Diversas outras configurações passaram então a ser testadas.

A hipótese da perda de pacotes ser causada pela redirecção da saída do tcpdump para outro *host* foi afastada pelos testes. Outras possíveis causas da perda de pacotes eram:

- O uso simultâneo de dois processos do tcpdump.
- O processamento simultâneo dos dados recolhidos pelo tcpdump.

Experimentou-se assim duas alternativas:

- Usar um único tcpdump (mantendo o processamento dos dados)
- Retirar o processamento dos dados (mantendo os dois processos do tcpdump)

Nenhuma das duas alternativas isoladamente conseguiu reduzir a perda de pacotes a níveis aceitáveis. Entretanto, a combinação das duas (usar um único tcpdump e retirar o processamento dos dados) conseguiu resolver o problema. Constatou-se que a capacidade de processamento do viriato era muito baixa. Sendo assim, a execução dos programas getcx e regista teria que ser feita em outro host.

Como o processamento dos dados é feito à medida em que os pacotes vão sendo recolhidos pelo tcpdump, o *pipe* foi mantido, desta vez fazendo a ligação entre programas executados em *hosts* diferentes. Para possibilitar isso, utilizou-se o *remote shell*. A configuração da linha de comando do sistema passou a ser a seguinte.

tcpdump <opções> <filtro> | rsh hermes -l <login> 'getcx | regista'

Para satisfazer às necessidades de recolha de tráfego de ambas as teses, o filtro utilizado para o tcpdump ficou mais complexo e passou-se a recolher muito mais pacotes do que seria necessário para esta implementação. O programa getcx também foi modificado para poder gravar em disco a informação que outrora era recolhida pela outra instância do tcpdump, relativa à outra tese. Com essas modificações, o sistema passou a funcionar com uma perda pequena de pacotes (inferior a 1 %).

Por diversas vezes, entretanto, após poucos dias de funcionamento ininterrupto a perda de pacotes voltava a atingir valores elevados, sem recuperar o estado anterior. A causa de tal comportamento não foi identificada.

Para contornar esse problema, o funcionamento do sistema é interrompido todo dia à mesma hora e imediatamente reiniciado, sendo a interrupção feita numa hora de tráfego reduzido. Com essa medida, a perda de pacotes nunca mais ultrapassou a marca de 1 %. Outro benefício desta medida é que se obtém diariamente a informação sobre o número de pacotes recebidos e perdidos, fornecidos pelo tcpdump no momento de sua interrupção, sendo bastante útil para o controlo do funcionamento do sistema.

Para a interrupção do processamento do sistema, utiliza-se o comando kill com a opção -2 (SIGINT), especificando-se o numero do processo corrente do tcpdump no viriato. Caso fosse utilizada a opção -9 (SIGKILL), a informação sobre o número de pacotes recebidos e perdidos não seria fornecida pelo tcpdump.

Este comando é colocado em uma *crontab*, seguido por outro que reinicializa o processamento. O sistema operativo encarrega-se de enviar um *mail* ao usuário a cada vez que o tcpdump é interrompido, contendo as mensagens direccionadas por este ao *stderr* (pois estas não seguem pelo *pipe*). O exemplo de um *mail* destes é mostrado a seguir:

```
From daemon Fri Sep 20 06:00:03 1996
Return-Path: <root>
Received: by viriato.rccn.net (4.1/SMI-4.1)
      id AA00696; Fri, 20 Sep 96 06:00:03 +0200
Date: Fri, 20 Sep 96 06:00:03 +0200
From: root (Operator)
Message-Id: <9609200400.AA00696@viriato.rccn.net>
To: root
Subject: Output from "cron" command
Status: RO

Your "cron" job

/usr/users/afonso/td.s

produced the following output:

tcpdump: listening on le0
tcpdump: Filtering in user process

7155267 packets received by filter
1804 packets dropped by kernel
```

Pensando em reduzir ainda mais a perda de pacotes e diminuir a quantidade de bytes transferidos do viriato para o hermes, resolveu-se dividir o processamento do tcpdump em duas partes, alterando-se a linha de comando para:

tcpdump -w- <opções> <filtro> | rsh hermes -l <login>
'tcpdump -r- <opções2> | getcx | regista'

Com a opção *-w-*, o tcpdump do viriato passa a fornecer a informação em bruto, um formato mais compacto, porém não legível. Medições indicam que esse formato reduz a quantidade de bytes produzidos em cerca de 64 %.

Por sua vez, o tcpdump do hermes, com a opção *-r-*, encarrega-se de mais uma parte do processamento anteriormente feito no viriato: a análise e conversão da informação para um formato legível. Medições indicam que o tempo de CPU gasto pelo tcpdump utilizado para recolha de pacotes é muito menor que o tempo gasto pelo tcpdump encarregado da conversão de formatos (cerca de 6 vezes). Com essa configuração, a perda de pacotes nunca ultrapassou 0,5 %, sendo em média inferior a 0,1 %.

A configuração utilizada atribui quase todo o processamento do sistema ao hermes. O trabalho de dividir o processamento entre dois *hosts* (e a consequente geração de tráfego entre ambos) podem ser evitados caso se utilize para a recolha de tráfego um *host* com capacidade suficiente para o processamento de todas as tarefas do sistema. Em testes, o viriato foi mais de 20 vezes mais lento que o hermes na execução das mesmas operações. Além disso, outro factor que contribui para a perda de pacotes no viriato é que, segundo o manual do tcpdump, o código de captura de pacotes do SunOS 4.1 não é eficiente.

5.2.2 Carga na Rede

A carga na rede varia de acordo com a hora do dia, tendo uma influência considerável nos tempos de resposta. A figura 10 apresenta a média móvel dos tempos de resposta apresentados na figura 3. O primeiro dia no gráfico é sexta-feira.

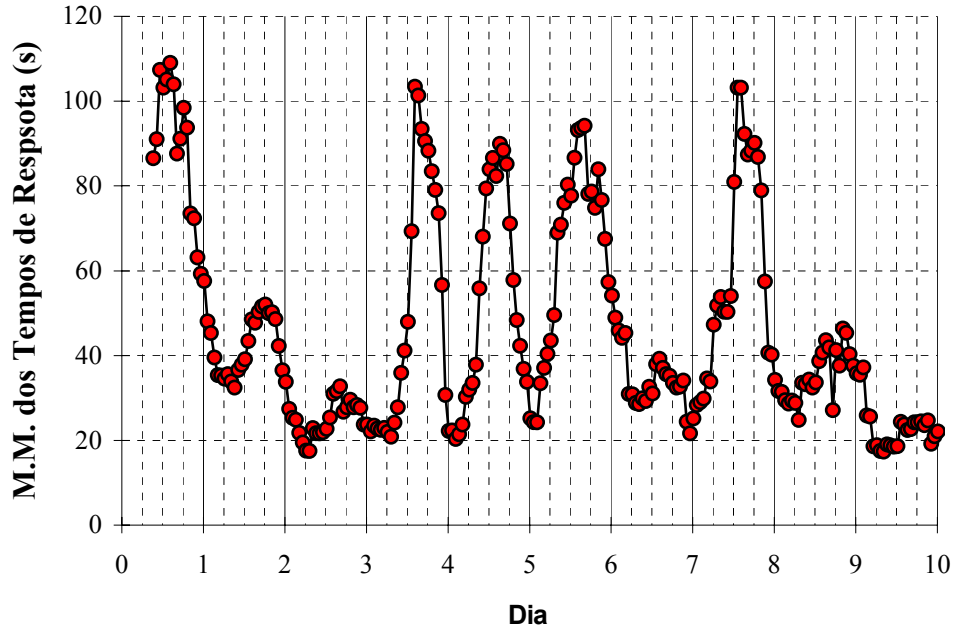


Figura 10: Média Móvel dos Tempos de Resposta de um Servidor na Internet ao Longo dos Dias.

O gráfico demonstra que o comportamento do tempo de resposta é sazonal. O seu valor tem tendência de aumentar nas horas de carga mais intensa na rede, e diminuir nas horas de menor tráfego. Nos fins de semana (dias 1, 2, 8 e 9) e feriados (dia 6), a carga na rede se reduz, consequentemente o tempo de resposta não atinge valores elevados como nos outros dias.

Os servidores usados para as medições apresentaram variações similares nas médias dos tempos de resposta ao longo do mesmo período, de acordo com medidas da correlação entre a médias móveis respectivas.. Os servidores mais próximos do cliente tendem a sofrer uma variação semelhante nos tempos de resposta, pela diferença de fuso horário ser pequena, mas mesmo o tempo de resposta do servidores mais afastados é influenciado pela carga nas redes próximas ao cliente.

O tempo de conexão também sofre a influência da carga na rede. Como as medições obtidas para os servidores são feitas em diferentes horas do dia, de acordo com os acessos dos clientes, as médias dos tempos de conexão dos servidores podem fornecer estimativas incorrectas. Para evitar isso, é feita a compensação do valor do tempo de conexão de acordo com uma estimativa da carga na rede no momento da medição.

A estimativa da carga na rede utilizada na implementação aproveita-se dos próprios tempos de conexão medidos. Seu valor é calculado com base na média geométrica (utilizando o *single exponential smoothing*) dos tempos de conexão de

todos os servidores, utilizando as equações 11 e 12, tal como é feito para cada servidor individualmente. Como a quantidade de amostras é muito maior, o coeficiente α utilizado é muito menor, para evitar variações muito bruscas na média. Para reduzir tendências no valor da estimativa causadas por seguidos acessos a um mesmo servidor em um curto período, o tempo de conexão só e considerado para o cálculo da média quando pertence a um servidor diferente do tempo considerado anteriormente. A estimativa é armazenada periodicamente em um ficheiro, fornecendo uma indicação da carga na rede ao longo do tempo. A figura 11 apresenta o valor desta estimativa ao longo do tempo para um período de cerca de uma semana.

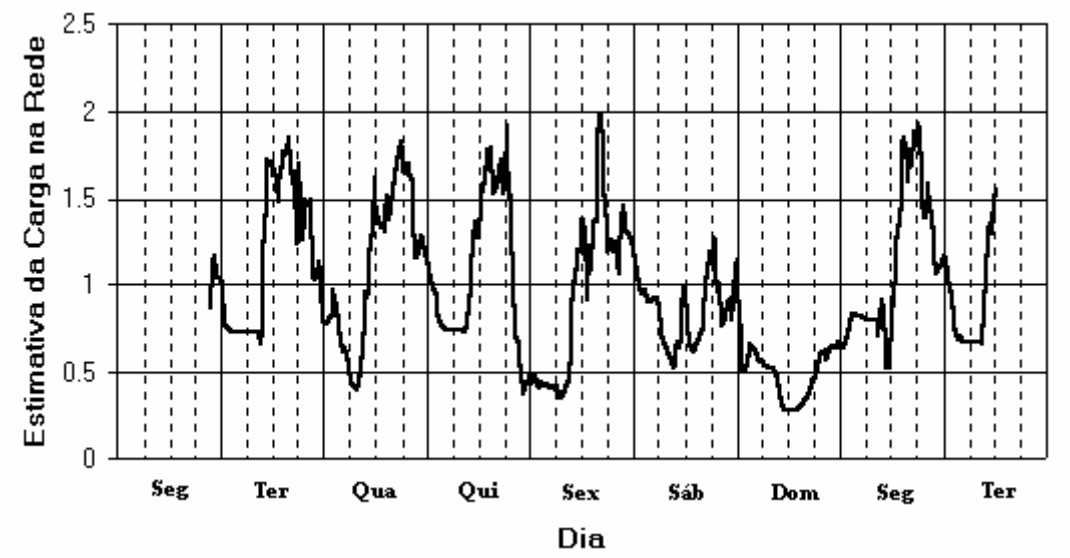


Figura 11: Estimativa da Carga na Rede ao Longo do Tempo.

O cálculo da compensação é feito dividindo-se o tempo de conexão do servidor (T_{cx_t}) pela estimativa da carga na rede (E_{cr_t}) naquele instante:

$$X_t = \frac{T_{cx_t}}{E_{cr_t}} \tag{13}$$

X_t é o novo valor da amostra no instante *t*, utilizado na equação 11, em substituição do tempo de conexão. A estimativa do tempo de resposta (E_{cx}), com base no tempo de conexão e na estimativa da carga na rede, é obtido da média geométrica (*GSt*) das amostras, calculada de acordo com as equações 10, 11 e 12.

$$E_{cx} = GS_t \tag{14}$$

5.3 Web Server Select

O Web Server Select [44] é um serviço criado para disponibilizar a informação contida na Tabela de QoS aos usuários via World-Wide Web. A página principal do serviço possui um *form* para a submissão da *query*, contendo uma janela de texto para que o utilizador introduza a lista de servidores de que se deseja obter informação. Também existe um campo em que o utilizador pode introduzir o valor do MSS utilizado pelo seu cliente.

Na submissão [45] do *form*, os dados são enviados a um programa que consulta a Tabela de QoS e produz uma nova página HTML com a resposta à *query*. A página apresenta uma tabela contendo uma linha para cada servidor identificado na Tabela de QoS, e as seguintes colunas com informação relativa a cada servidor:

- Nome de domínio.
- Endereço IP.
- Estimativa da Qualidade de Serviço (E_{QoS}).
- Estimativa do tempo de resposta (E_{cx}), baseada na média dos tempos de conexão, compensados pela estimativa da carga na rede.
- MSS do servidor (MSS_s).
- Número de pedidos de conexão (N_{req}).
- Número de respostas afirmativas a pedidos de conexão (N_{resp}).
- Estimativa da disponibilidade recente (D_{rec}).
- Data do último acesso ao servidor.

A tabela é ordenada pela estimativa da qualidade de serviço (E_{QoS}), que é calculada com base na estimativa do tempo de resposta (E_{tr}) e da disponibilidade recente (D_{rec}):

$$E_{QoS} = \frac{E_{tr}}{D_{rec}} \tag{15}$$

A estimativa do tempo de resposta, por sua vez, corresponde à estimativa feita com base nos tempos de conexão (E_{cx}), compensada pelo MSS da conexão.

$$E_{tr} = k\, E_{cx}\, F_{MSS} \tag{16}$$

Em que k é uma constante e F_{MSS} é o factor de compensação do MSS definido na equação 5.

As estimativas calculadas neste capítulo não possuem nenhuma unidade em particular, e não se pretende utilizá-las para prever a demora na obtenção de um

determinado recurso por parte de um servidor, mas sim para a comparação entre servidores.

5.3.1 Exemplo de Utilização

A seguir é apresentado um breve exemplo de utilização do Web Server Select. A figura 12 mostra a página inicial do serviço, em que existe uma área de texto no qual é introduzida a lista de servidores desejada. No exemplo em questão utilizou-se alguns dos *mirrors* existentes para o *site* da Tucows.

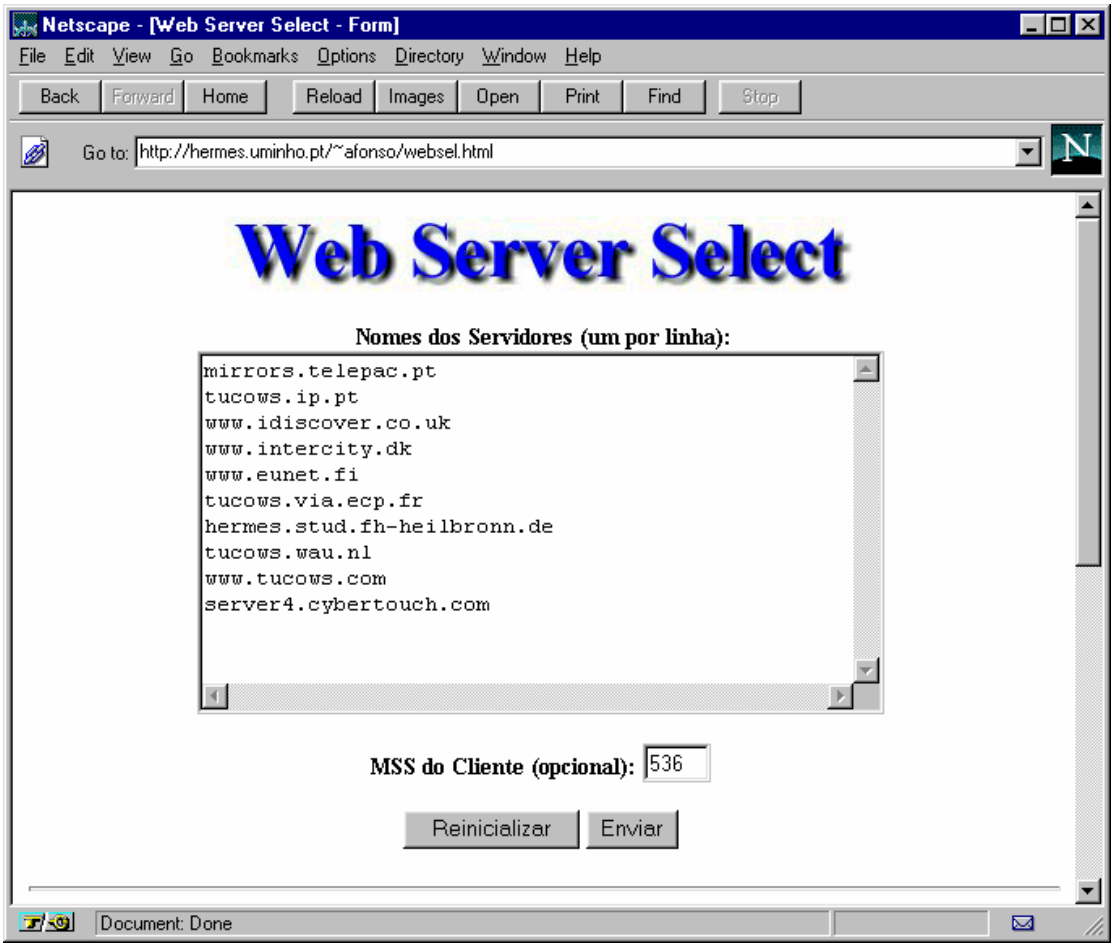


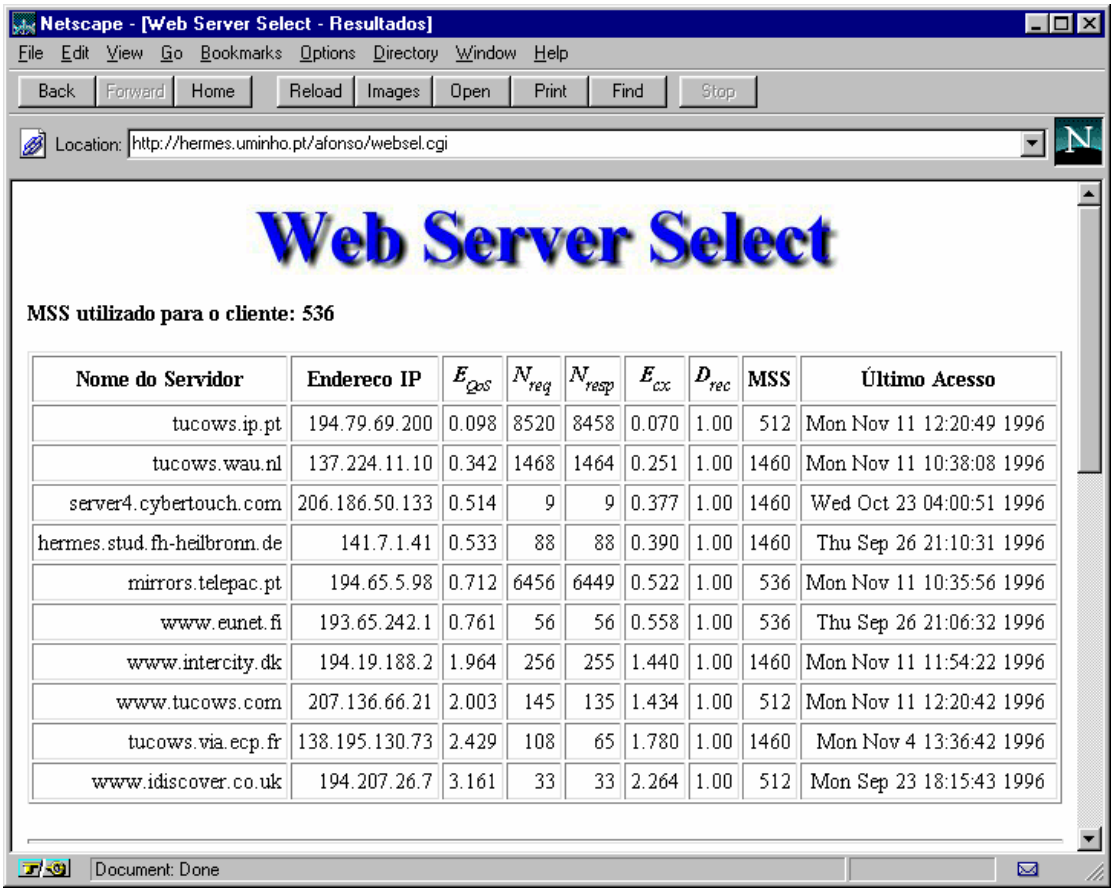
Figura 12: Página Inicial do Web Server Select Apresentando o Form Preenchido com uma Query Exemplo.

No exemplo são utilizados dez servidores, sendo dois em Portugal, seis no resto da Europa, um no Canadá e um nos Estados Unidos (o servidor principal, www.tucows.com). O MSS usado para o cliente neste caso é 536, que corresponde ao valor *default*.

A figura 13 apresenta os resultados obtidos com essa *query*. O servidor escolhido (tucows.ip.pt) localiza-se em Portugal e apresenta uma estimativa bastante inferior aos outros servidores. Já o outro servidor de Portugal (mirrors.telepac.pt) ocupa uma posição intermediária. O servidor do Canadá (server4.cybertouch.com) ocupa uma boa posição, embora o número de amostras disponíveis seja pequeno. Isso aliado ao facto das amostras mais recentes de alguns servidores terem mais de um mês de idade reforça a necessidade da utilização do maior número de clientes possíveis na recolha de tráfego.

O servidor principal (www.tucows.com) é um dos que apresenta pior estimativa, o que indica que os *mirror sites* são de muita utilidade para se obter os recursos disponibilizados num tempo muito menor. Isso sem contar que os mirrors ajudam a distribuir a carga, pois se o site principal atendesse aos pedidos de todos os clientes seus tempos de resposta seriam muito mais elevados.

A estimativa da disponibilidade recente (D_{rec}) para todos os servidores no exemplo apresenta o valor máximo, significando que todos servidores estavam disponíveis da última vez em que foram acedidos, embora alguns servidores apresentem o número de respostas (N_{resp}) inferior ao número de pedidos (N_{req}), denunciando a falha em responder a alguns pedidos durante o período de recolha de tráfego.



Nome do Servidor	Endereco IP	E_{QoS}	N_{req}	N_{resp}	E_{cx}	D_{rec}	MSS	Último Acesso
tucows.ip.pt	194.79.69.200	0.098	8520	8458	0.070	1.00	512	Mon Nov 11 12:20:49 1996
tucows.wau.nl	137.224.11.10	0.342	1468	1464	0.251	1.00	1460	Mon Nov 11 10:38:08 1996
server4.cybertouch.com	206.186.50.133	0.514	9	9	0.377	1.00	1460	Wed Oct 23 04:00:51 1996
hermes.stud.fh-heilbronn.de	141.7.1.41	0.533	88	88	0.390	1.00	1460	Thu Sep 26 21:10:31 1996
mirrors.telepac.pt	194.65.5.98	0.712	6456	6449	0.522	1.00	536	Mon Nov 11 10:35:56 1996
www.eunet.fi	193.65.242.1	0.761	56	56	0.558	1.00	536	Thu Sep 26 21:06:32 1996
www.intercity.dk	194.19.188.2	1.964	256	255	1.440	1.00	1460	Mon Nov 11 11:54:22 1996
www.tucows.com	207.136.66.21	2.003	145	135	1.434	1.00	512	Mon Nov 11 12:20:42 1996
tucows.via.ecp.fr	138.195.130.73	2.429	108	65	1.780	1.00	1460	Mon Nov 4 13:36:42 1996
www.idiscover.co.uk	194.207.26.7	3.161	33	33	2.264	1.00	512	Mon Sep 23 18:15:43 1996

Figura 13: Resultado da Query Exemplo ao Web Server Select.

Muitas vezes listas de servidores aparecem disponíveis em páginas HTML. Por isso, pensa-se em incluir no Web Server Select uma opção que possibilite ao usuário a submissão de um documento HTML, em vez de uma lista de servidores. O programa se encarregaria da tarefa de percorrer o documento e extrair deste a lista de servidores.

5.4 Expansão a Outros Clientes

Quanto maior a quantidade de acessos a servidores disponíveis, mais completa e actualizada estará a Tabela de QoS; portanto, é desejável englobar um grande número de clientes. Na implementação actual são utilizados clientes de três redes locais ao campus de Gualtar da Universidade do Minho. Pode-se aumentar o número de clientes disponíveis simplesmente englobando outras redes do campus.

Para aumentar ainda mais o número de clientes pode-se cogitar a utilização de outras redes da Universidade do Minho, externas ao campus de Gualtar. Para isso, deve-se avaliar se a distância entre o cliente e o ponto de recolha do tráfego pode afectar as medições.

Para o caso da métrica utilizada na implementação, ou seja, o tempo de estabelecimento de conexão; a localização do cliente não terá grande influência no resultado da medição, pois o que se mede é o tempo desde a passagem do pedido de estabelecimento de conexão até a passagem da resposta do servidor, ou seja, é como se o cliente estivesse localizado nesse ponto. Portanto, a princípio, todos os clientes internos à universidade fornecem uma medida uniforme do tempo de conexão para um mesmo servidor.

Já para o caso da métrica *b*, definida na secção 4.3, os resultados seriam bastante influenciados pela localização do cliente. O tempo de resposta utilizado na métrica provém da medição do tempo decorrido deste a detecção do pedido de estabelecimento de conexão até a detecção pedido de término da conexão, ambos da parte do cliente. A partir da figura 1, pode-se observar que não há alteração significativa no tempo medido, qualquer que seja a posição do ponto de recolha de tráfego no caminho entre o cliente e o servidor, sendo basicamente o mesmo tempo visto a partir do cliente. Caso essa métrica fosse utilizada, os clientes forneceriam medidas do tempo de resposta (de um mesmo servidor) que seriam dependentes da sua distância em relação ao ponto de recolha do tráfego.

Existe uma pequena deficiência no processo de identificação de conexões implementado actualmente (baseado na métrica *a* da secção 4.3), mas que pode se tornar relevante caso se deseje incluir clientes afastados na recolha de tráfego. Quando é identificado um pedido de estabelecimento de uma conexão, seus dados são colocados provisoriamente em uma tabela. Pedidos resultantes de retransmissões são ignorados após terem sido comparados com as conexões em aberto existentes na tabela. Quando chega a resposta a um pedido de conexão, os dados referentes à conexão são enviados à etapa seguinte no processamento, e as referências à conexão são removidas da tabela. Nesse ponto, se aparecer um pedido de estabelecimento de conexão referente a uma conexão já efectuada, será interpretado como se fosse referente a uma nova conexão, causando um erro (figura 14).

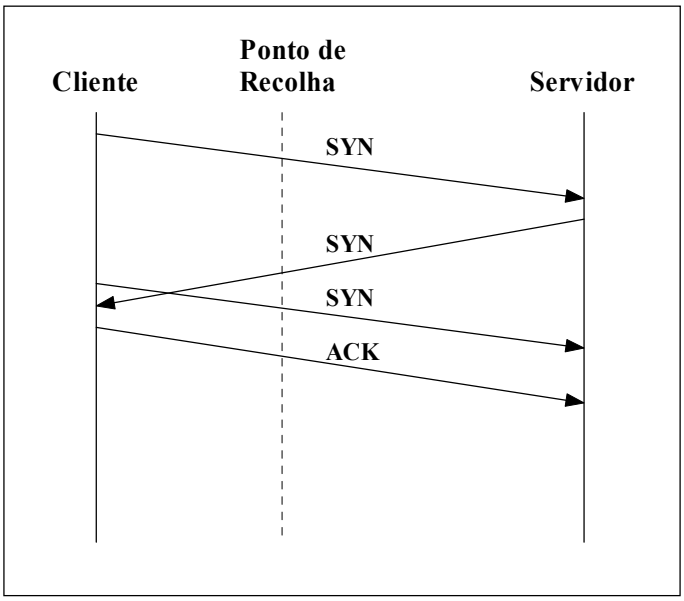


Figura 14: Identificação Equivocada de uma Nova Conexão.

Depois que o cliente recebe a resposta, não repete mais o mesmo pedido, mas como o ponto de recolha de tráfego localiza-se entre o cliente e o servidor, pode ser que a retransmissão se efectue imediatamente antes da chegada da resposta ao cliente, e chegue ao ponto de recolha após a passagem do pacote de resposta. Este evento é improvável quando o ponto de recolha está próximo dos clientes, como é o caso na implementação actual, mas pode acontecer com alguma frequência se forem utilizados clientes mais afastados. Portanto, para a utilização destes clientes torna-se necessária uma pequena modificação no programa encarregado da identificação das conexões, para que rejeite pacotes repetidos vindos do cliente durante um certo período após o estabelecimento da conexão.

Na implementação actual os servidores internos à Universidade do Minho não são incluídos na Tabela de QoS, porque os pedidos feitos pelos clientes internos não passam pelo ponto de recolha de tráfego. Pode-se então pensar em utilizar os acessos feitos pelos clientes externos para fornecer informação sobre os servidores internos. Uma diferença daí resultante é que o número de pedidos registados na Tabela de QoS para esses servidores seriam referentes aos clientes externos, não aos clientes internos. Como foi visto anteriormente, a estimativa depende basicamente do ponto de recolha do tráfego, e não da localização do cliente, pois o tempo de conexão é utilizado como métrica. Assim, é como se a estimativa fosse feita utilizando-se um cliente local, situado no ponto de recolha.

6 Conclusões

6.1 Prós e Contras do Método Utilizado

Dentre os métodos de selecção dinâmicos, o método idealizado neste trabalho apresenta vantagens importantes em relação aos métodos temporais normalmente usados.

- A principal vantagem é evitar totalmente a inserção de tráfego na rede para o cálculo das estimativas, por se utilizar a técnica de *passive probing*. Entre os outros métodos, há alguns que geram pouco tráfego, mas conseguem estimativas grosseiras, e outros que conseguem estimativas mais rigorosas ao custo de produzirem uma quantidade apreciável de tráfego adicional. O custo da obtenção das estimativas é agravado por se fazer o *probing* a diversos servidores de cada vez.

- Este método fornece resposta imediata às *queries*, pois as estimativas são armazenadas na Tabela de QoS para utilização quando é necessário. Os outros métodos normalmente são reactivos, significando que o processo de obtenção das estimativas só é iniciado após a *query* ter sido recebida. O usuário terá que esperar o tempo necessário à sua obtenção, sendo que o uso de uma estimativa mais rigorosa e complexa tende ainda a aumentar esse tempo.

- Por utilizar no cálculo das estimativas medições relativas ao protocolo com o qual se deseja realmente aceder ao servidor, este método tem a tendência de fornecer estimativas mais confiáveis em certos aspectos, como a carga e a disponibilidade do servidor. Isso em oposição a outros métodos que normalmente utilizam as mensagens de *echo* do ICMP, e em alguns casos o UDP.

- A Tabela de QoS também armazena os nomes e endereços IP dos servidores. Desta forma, o cliente pode fazer a *query* utilizando nomes, se lhe convier, em vez de ter que obter os endereços de todos os servidores antes da selecção. Todas as *queries* ao DNS podem ser evitadas, inclusive a *query* ao servidor escolhido, poupando tempo e recursos.

A técnica de *passive probing* possui algumas desvantagens, entretanto. Sendo assim, o método proposto neste trabalho não está isento de problemas. Porém, na maior parte dos casos foram encontradas soluções satisfatórias.

- A carga na rede é variável com o tempo, influenciando as medições. Como não existe controlo sobre o momento em que são feitas as medições, pode haver influência sobre as estimativas dos servidores. Para minimizar essa influência, utiliza-se uma estimativa da carga na rede para compensar os seus efeitos sobre as medições.

- Existe pouca liberdade na escolha da métrica a utilizar no cálculo das estimativas, estando-se restrito ao processo de comunicação habitual entre clientes e servidores, enquanto com *active probing* pode-se gerar o tráfego da maneira desejada. Apesar disso, a métrica escolhida (tempo de conexão) produziu resultados bastante satisfatórios para o propósito da selecção.

- As medições são realizadas por diferentes clientes, cujo comportamento pode influenciar na obtenção das estimativas. Caso fosse utilizada a métrica baseada no tempo de resposta, por exemplo, um cliente com problemas poderia causar atrasos na comunicação, e o resultado se reflectiria na estimativa do servidor. Com o uso dos tempos de conexão, porém, a influência do cliente torna-se bem menor. Esta métrica também é relativamente independente da distância entre o cliente e o ponto de recolha de tráfego, ao contrário do que acontece com a métrica baseada no tempo de resposta.

- Não há controlo sobre o número de acessos feitos aos servidores. Alguns servidores terão poucos acessos, e outros poderão não ter informação disponível na Tabela de QoS, por nenhum cliente local tê-los contactado até aquele momento. Esta probabilidade, porém, é maior para servidores distantes e desconhecidos, que provavelmente não serão boas opções de selecção, de qualquer forma. Portanto, a informação relativa a esses servidores tende a ser de menor importância.

6.2 Sobre os Resultados

Os resultados apresentados na secção 4.5.1 mostram que o *ranking* dos servidores baseado nos tempos de resposta médios tende a manter-se razoavelmente estável num período de muitos dias, apesar de os tempos de resposta quando vistos isoladamente apresentarem variações muito bruscas. Isso permite que uma estimativa baseada na média entre medições distribuídas ao longo do tempo, como a utilizada nesta tese, possa ter sucesso.

O objectivo principal definido anteriormente foi alcançado, de acordo com os resultados da secção 4.5, pois conseguiu-se obter uma redução no tempo médio de resposta. Além disso, as outras características desejáveis ao método de selecção também foram satisfeitas.

Os métodos de selecção dinâmicos, em particular os métodos temporais, mostraram, ao longo dos capítulos 3 e 4, clara vantagem na redução do tempo de resposta e distribuição mais eficiente do tráfego na rede, em relação aos métodos estáticos, como os baseados na distância geográfica ou no número de *hops*, desaconselhando a utilização destes últimos.

A utilização do tempo de conexão mostrou vantagens em relação ao uso do *round-trip time* medido com o ping, por fornecer medidas mais correctas tanto da carga quanto da disponibilidade do servidor. Além disso, ele é apropriado à utilização com o *passive probing*, e fornece estimativas baseadas na média que apresentam melhores resultados, por sofrer a influência da perda de pacotes e retransmissão, considerando assim os efeitos de congestão na rede com maior eficiência.

Métodos de selecção baseados na estimativa pela média podem ter mais êxito do que alguns métodos baseados na estimativa instantânea, principalmente quando estes últimos são muito simples, como mostram os resultados na secção 4.5. O método de selecção baseado na estimativa instantânea, utilizando como métrica uma única medição do tempo de conexão (similar à utilização do *round-trip time*), teve como resultado um maior tempo de resposta médio do que o método de selecção baseado na média dos tempos de conexão, nos casos analisados.

O MSS utilizado na conexão mostrou ter influência no tempo de resposta, justificando a sua utilização no cálculo da estimativa. Entretanto, é provável que a sua influência tenda a diminuir com o aumento no tamanho do documento transferido. Neste caso, convém incluir o tamanho do documento na equação 5, que define o factor de compensação do MSS, de forma a tornar a estimativa mais acurada.

Os resultados experimentais obtidos foram bons, mas é desejável realizar um conjunto maior de medições para poder comprovar a eficiência do método utilizado sob outras condições, e verificar se existe a necessidade de alterações ou adaptações. Para isso convém a realização mais testes, com outros servidores, tamanhos de documentos e clientes, inclusive de fora da Universidade do Minho.

6.3 Trabalho Futuro

6.3.1 Automatização do Processo

Existem basicamente três etapas desde a identificação do recurso desejado pelo usuário até a sua obtenção:

- Localização do recurso
- Selecção de um dos servidores
- Obtenção do recurso no servidor escolhido

Actualmente é utilizado o URL para identificar o recurso no WWW. Sendo assim, a etapa de localização do recurso não é necessária, porque o URL já localiza o recurso. A selecção não faz sentido, porque não há opções, só resta a obtenção do

recurso, que é feita automaticamente, mas deixando de lado outros servidores que poderiam fornecer o recurso com uma qualidade de serviço superior.

Para que os usuários possam usufruir plenamente do método de selecção apresentado nesta tese, é necessária a automatização do processo, o que não depende apenas da etapa de selecção. É indispensável a utilização de um processo automático também para a localização do recurso, tal como a resolução URN2URC, que é alvo de investigação actualmente, existindo várias propostas de implementação.

Escolhido um recurso pelo usuário, através de um identificador (URN), o processo de resolução conduz à obtenção automática de uma lista de URLs que apontam para o recurso. Neste ponto estão criadas condições para se processar a selecção seguida pela obtenção do recurso de forma automática, ou seja, sem necessidade de intervenção por parte do usuário.

Pode-se modificar um cliente para que este aceda à Tabela de QoS, obtenha a informação do servidor mais apropriado, e o contacte. Para que a Tabela de QoS esteja disponível a mais de um cliente, deve-se ainda implementar um protocolo como o descrito na secção 4.6.2.

Em vez disso pode-se optar por modificar um servidor *proxy*, para que seja ele a atender os pedidos por parte dos clientes e aceder à Tabela de QoS, dispensando modificações nos cliente, bem como a utilização de um protocolo apropriado para o acesso à Tabela de QoS.

Ou pode-se fazer ambas as coisas, pois as alternativas podem coexistir.

6.3.2 Extensão a Outros Protocolos

O serviço de selecção entre servidores foi implementado inicialmente para o HTTP, o que não significa que não possa ser ampliado para englobar outros protocolos.

Para estender a colecta de informação aos servidores FTP, basta incluir a porta 21 do TCP no filtro do tcpdump. Como muitos hosts abrigam tanto servidores de HTTP como de FTP e outros protocolos, uma questão que se põe é se a informação relativa aos tempos de conexão e demais campos na tabela de QoS deve ser organizada por host ou por par <host, protocolo>. Caso o tempo de conexão para um host seja independente do protocolo, como parece ser o caso, o uso de uma média única dos tempos de conexão para cada host é mais vantajosa, pois aumenta o número de amostras disponíveis para os servidores em geral.

Quanto à disponibilidade de um servidor, esta dependerá em grande parte da disponibilidade do host e da rede, que influenciam igualmente todos os servidores em um host, de forma que o uso de uma única estimativa de disponibilidade para todos os protocolos pode vir a funcionar na maior parte dos casos. Quanto à informação do número de acessos feitos ao host, parece mais interessante fazer a distinção entre os protocolos. De uma forma ou de outra, a extensão da tabela de QoS a outros protocolos não apresenta grandes obstáculos em termos de implementação.

Pode-se também considerar o uso da informação disponível na tabela de QoS para a selecção entre servidores de outros protocolos não incluídos na colecta de tráfego, sejam esses servidores baseados no TCP, no UDP ou outros protocolos. A distância entre os hosts, a carga na rede e no host acabam por influenciar todos os

protocolos, portanto a estimativa baseada em medições realizadas para um protocolo podem servir como aproximação para outros. Essa premissa é considerada pela maioria dos trabalhos relacionados com a selecção entre servidores, nos quais normalmente não se faz a distinção do protocolo para o qual a estimativa de proximidade se destina.

6.4 Observações Finais

Em [36] os autores referem a construção de uma versão de um browser WWW que utiliza as ferramentas desenvolvidas em seu trabalho para informar ao usuário sobre a estimativa da velocidade de transferência para cada *link* de hipertexto existente nas páginas WWW. Os *links* seriam coloridos com uma medida do estado corrente da rede; por exemplo, os *links* mais rápidos seriam coloridos a verde, enquanto os mais lentos a vermelho. Essa ideia também pode ser utilizada com este trabalho, com a vantagem de que as estimativas já estão disponíveis a priori.

Os dados armazenados na tabela de QoS podem ser úteis para outros propósitos além da selecção do servidor. A informação sobre os servidores com mais acessos localmente, os servidores com melhores e piores tempos de resposta, a distribuição dos acessos por domínios e outras estatísticas podem ser obtidas da Tabela de QoS.

O tempo necessário à resolução do nome do servidor para o endereço IP respectivo não foi incluído na estimativa do tempo de resposta (diferentemente do que foi feito em [34]). O motivo é que esse tempo não se relaciona directamente com a qualidade de serviço oferecida pelo servidor, mas sim com o serviço de DNS, que por sua vez é bastante influenciado pela existência ou não da informação requisitada em *cache*. Nos casos em que a informação não se encontra disponível localmente em *cache* (tipicamente quando se deseja contactar um servidor que não tenha sido acedido recentemente) pode ser necessário o contacto com diversos servidores de nomes até que se obtenha o endereço pretendido, o que pode demorar muito tempo. E como a Tabela de QoS armazena tanto o nome quanto o endereço IP do servidor, o tempo necessário à resolução do endereço IP do servidor pode ser suprimido.

O coeficiente α utilizado na equação 11 para o cálculo da média dos tempos de conexão é mantido constante após obtido um número razoável de amostras. Pode-se considerar a utilização de um coeficiente α variável, com o objectivo de procurar aumentar a acurácia das estimativas em alguns casos. Uma hipótese consiste em fazer α variar em função do tempo decorrido entre a obtenção da última amostra e a amostra actual; e outra hipótese consiste em fazer o valor de α proporcional ao valor do erro da última estimativa.

O tempo de resposta é afectado por diversos factores (latência, carga no servidor, largura de banda disponível, porcentagem de perda de pacotes, etc), e a realização de um grande número de medições para o cálculo de estimativas em separado de cada um desses factores pode trazer custos muito elevados. Isso sem contar que tempo de resposta também é influenciado pela actuação do cliente, a versão do protocolo de transporte utilizado por cada uma das partes, o protocolo de nível aplicação e sua versão, etc. Aliando isso à alta variabilidade das medições torna-se difícil atribuir um peso adequado para cada uma das estimativas parciais no cálculo

da estimativa do tempo de resposta, bem como a comprovação da correcção das estimativas. O uso da média dos tempos de estabelecimento de conexão não pretende fornecer uma estimativa rigorosa do tempo de resposta, mas sim proporcionar uma estimativa simples, de baixo custo, flexível, mas ao mesmo tempo eficiente. Para o propósito da selecção entre servidores a estimativa parece adequar-se bastante bem.

Referências

[1] Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen and Arthur Secret, “The World-Wide Web”, Communications of the ACM, Vol. 37, No. 8, pp 76-82, Agosto 1994.

[2] James D. Guyton and Michael F. Schwartz, “Locating Nearby Copies of Replicated Internet Servers”, Technical Report CU-CS-762-95, Department of Computer Science, University of Colorado, Boulder, USA., Fevereiro 1995.

[3] Comer, Douglas E., “Internetworking With TCP/IP - Vol I: Principles, Protocols and Architecture”, Prentice-Hall International, 1990.

[4] J. Postel (ed.), "Transmission Control Protocol - DARPA Internet Program Protocol Specification", RFC 793, Setembro 1981.

[5] T. Berners-Lee, L. Masinter, M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, Dezembro 1994.

[6] D. Raggett, "HyperText Markup Language Specification Version 3.0", Internet-Draft (draft-ietf-html-specv3-00.txt), Work in Progress from the HTML Working Group of the IETF, Março 1995.

[7] W3C, "HTML Specs, Drafts and Reports", 1996.
URL: http://www.w3.org/pub/WWW/Markup/Bibliography.

[8] Berners-Lee, T., Fielding, R., Frystyk, H., "Hypertext Transfer Protocol - HTTP/1.0", RFC 1945, Maio 1996.

[9] W3C, "HTTP - Hypertext Transfer Protocol", 1996.
URL: http://www.w3.org/pub/WWW/Protocols/.

[10] "External Traffic by Protocol", University of Waterloo, Canada, Maio 1996.
URL: http://www.dcs.uwaterloo.ca/cn/Stats/ext-prot.html.

[11] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk and T. Berners-Lee, “Hypertext Transfer Protocol - HTTP/1.1”, Internet-Draft (draft-ietf-http-v11-spec-07.txt), Work in Progress from the HTTP Working Group of the IETF, Agosto 1996.

[12] Luotonen, A., Altis, K., “World-Wide Web Proxies”, Computer Networks and ISDN Systems, First International Conference on The World-Wide Web, Elsevier Science BV, Abril 1994.

[13] Chankhunthod, A., Danzig, P. B., Neerdaels, C., Schwartz, M. F., Worrell, K. J., “A Hierarchical Internet Object Cache”, Technical Report CU-CS-766-95,

Department of Computer Science, University of Colorado - Boulder, USA, Março 1995.

[14] James S. Gwertzman and Margo Seltzer, “The Case for Geographical Push-Caching”, Technical Report HU TR-34-94, Harvard University, DAS, Cambridge, MA, USA, 1994.

[15] Azer Bestavros, “Demand-based Document Dissemination for the World-Wide Web”, Technical Report 95-003, Computer Science Department, Boston University, Fevereiro 1995.

[16] K. Sollins, L. Masinter, "Functional Requirements for Uniform Resource Names", RFC 1737, Dezembro1994.

[17] M. Mealling, P. Falstrom, L. L. Daigle, "Uniform Resource Names, ISO OIDs and DNS", Internet-Draft (draft-mealling-oid-dns-00.txt), Work in Progress from the IETF, Novembro 1995.

[18] R. Daniel, M. Mealling, "URC Scenarios and Requirements", Internet-Draft (draft-ietf-uri-urc-req-00.txt), working in Progress from the IETF, Novembro 1994.

[19] L. Girod, K. Sollins, "Requirements for URN Resolution Systems", Internet-Draft (draft-girod-urn-res-require-00.txt), Work in Progress from the IETF, Junho 1996.

[20] Computerlink Online Inc., "Tucows - The Ultimate Collection of Winsock Software".
URL: http://www.tucows.com/.

[21] "The Linux Documentation Project Home Page".
URL: http://ftp.dei.uc.pt/mdw/linux.html.

[22] Martin Koster, "ArchiePlexForm".
URL: http://s700.uminho.pt/CGI/archieplex/archieplexform.html.

[23] CNET Inc., "SHAREWARE.COM".
URL: http://www.shareqare.com/.

[24] P. Mockapetris, "Domain Names - Implementation and Specification", RFC 1035 Novembro 1987.

[25] P. Mockapetris, "Domain Names - Concepts and Facilities", RFC 1034, Novembro 1987.

[26] J. Postel, J. Reynolds, "File Transfer Protocol (FTP)", RFC 959, Outubro 1995.

[27] Netscape Communications Corporation, "Netscape Navigator Family".

URL: http://home.netscape.com/comprod/products/navigator/index.html.

[28] Netscape Communications Corporation, “FTP’ing through a firewall (passive FTP)”, Maio 1996.
URL: http://help.netscape.com/kb/client/960513-36.html.

[29] S. McCanne, C. Leres, Van Jacobson, "Tcpdump Software", Julho 1996.
URL: ftp://ftp.ee.lbl.gov/tcpdump.tar.Z.

[30] C. Farrel, M. Schulze, S. Pleitner, D. Baldoni, "DNS Encoding of Geographical Location", RFC1712, Novembro1994.

[31] James Gwertzman, “Autonomous Replication in Wide-Area Internetworks”, Thesis to Computer Science for degree of Bachelor of Arts, Harvard College, Cambridge, Massachusetts, USA, Abril 1995.

[32] Van Jacobson, "Traceroute Software", Setembro 1996.
URL: ftp://ftp.ee.lbl.gov/traceroute.tar.Z.

[33] Mark E. Crovella and Robert L. Carter, “Dynamic Server Selection in the Internet”, Technical Report TR-95-014, Computer Science Department, Boston University, USA., Junho 1995.

[34] Charles L. Viles and James C. French, “Availability and Latency of World Wide Web Information Servers”, Department of Computer Science, University of Virginia, 1994.

[35] Jason Cox, Stan Green, Keith Moore, "SONAR - A Network Proximity Service", Internet-Draft (draft-moore-sonar-01.txt), Work in Progress from the Network Working Group of the IETF, Fevereiro 1996.

[36] Robert L. Carter and Mark E. Crovella, "Dynamic Server Selection using Bandwidth Probing in Wide-Area Networks", Technical Report BU-CS-06-007, Computer Science Department, Boston University, USA., Março 1996.

[37] Paul Barker, "Providing the X.500 Directory User with QoS Information", Computer Communication Review, ACM SIGCOMM, pp 28-37, 1994.

[38] Rio, M., Macedo, J., Costa, A., Freitas, V., "Supporting a URI infrastructure by Message Broadcasting", Proceedings INET '95 - Internet Society's 1995 International Networking Conference, Honolulu, Hawaii, USA, 1995.

[39] R. Enger, J. Reynolds, "FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices", RFC1470, Junho 1993.

[40] Mogul, J. C., "The Case for Persistent-Connection HTTP", ACM SIGCOMM’95 Conference, Cambridge, Massachusetts, USA:, 1995.

[41] "RCUM: Rede de Comunicações da Universidade do Minho".
URL: http://www.ci.uminho.pt/nucleos/ComCIUM/RCUM/rcum.html.

[42] B. Kernighan, D. Ritchie, "The C Programming Language", Prentice Hall, 1979.

[43] Makidrakis, S., WeelWrigth, S., McGee, V., "Forecasting Methods and Applications", Jon Wiley & Sons, pp 84-111, 1983.

[44] J. Afonso, "Web Server Select", Departamento de Informática, Universidade do Minho.
URL: http://hermes.uminho.pt/~afonso/websel.html.

[45] D. R. T. Robinson, “The WWW Common Gateway Interface Version 1.1”, Internet-Draft (draft-robinson-www-interface-01.txt), Work in Progress from the IETF, Fevereiro 1996.

[46] Van Paxson, "Empirically-Derived Analytic Models of Wide-Area TCP Connections", IEEE/ACM Transactions on Networking, 2(4):316-336, Agosto 1994.